Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Miha Zorec

Implementacija brezstičnih pametnih kartic na napravah NFC

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Viljan Mahnič

Ljubljana, 2014

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Implementacija brezstičnih pametnih kartic na napravah NFC

Tematika naloge:

Proučite delovanje brezstičnih pametnih kartic (komunikacijski protokol, razporeditev datotek na kartici, zagotavljanje varnosti) in različne izvedbe, ki so trenutno na razpolago (komercialne in odprtokodne). Nato predstavite možnosti za simulacijo fizičnih kartic z navideznimi karticami na pametnih mobilnih napravah. Pri tem poiščite najprimernejšo rešitev za realizacijo varnostnega elementa in predstavite model uporabe navideznih brezstičnih pametnih kartic, ki bo podpiral kartice različnih ponudnikov storitev.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Miha Zorec, z vpisno številko **63030184**, sem avtor diplomskega dela z naslovom:

Implementacija brezstičnih pametnih kartic na napravah NFC

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Viljana Mahniča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15. septembra 2014 Podpis avtorja:

Zahvalil bi se Petru Brajaku za idejo in pomoč pri izdelavi idejne rešitve. Zahvalil bi se ženi, ki me je prepričala, da končam študij. Zahvalil bi se za potrpežljivost mojima staršema, ki sta z malce zamude le dočakala mojo diplomo. Pa tudi za vso pomoč in podporo v času študija. Zahvalil bi se tudi izr. prof. dr. Viljanu Mahniču za mentorstvo.

Moji Ani Isabel.

Kazalo

Seznam uporabljenih kratic

3DES Triple DES. Trojni DES algoritem za šifriranje.

AES Advanced Encryption Standard. Napredni simetrični algoritem za kriptiranje.

AID Application identifier. Identifikator aplikacije, namenjen naslavljanju te aplikacije.

APDU Application protocol data unit. Komunikacijski protokol za prenos podatkov med pametno kartico in bralnikom pametnih kartic.

API Application programming interface. Programski vmesnik.

DES Data Encryption Standard. Simetrični algoritem za šifriranje.

EEPROM Electrically erasable programmable read-only memory Električno izbrisljiv programirljiv bralni pomnilnik.

HTML Hypertext markup language. Označevalni jezik za oblikovanje večpredstavnostnih dokumentov.

ICCID Integrated circuit card identifier. Vgrajeni unikatni identifikator SIM kartice.

JVM Java Virtual Machine. Java navidezna naprava.

MicroSD Bliskovni pomnilnik majhnih dimenzij.

MNO Mobile network operator. Mobilni operater.

NFC Near field communication. Nabor standardov za brezstično komunikacijo mobilnih naprav na kratki razdalji.

OTA Over the air secure channel. Varni kanal za prenos podatkov po zraku.

PIN Personal identification number. Osebna identifikacijska številka. Krajša številka namenjena zaščiti pred nepooblaščeno uporabo.

RAM Random-access memory. Bralno-pisalni pomnilnik.

REST Representational state transfer. Protokol za prenos podatkov preko interneta.

RFID Radio-frequency identification. Uporaba elektromagnetnega polja za brezžičen prenos podatkov.

ROM Read-only memory. Bralni pomnilnik.

SIM Subscriber identification module. Kartica za identifikacijo uporabnika v mobilnem omrežju.

SOAP Simple object access protocol. Protokol za spletne storitve, ki temelji na XML.

TSM Trusted service manager. Zaupanja vreden upravljavec storitev.

USB Universal Serial Bus. Univerzalno serijsko vodilo.

WI-FI Brezžična povezava elektronskih naprav.

Povzetek

Vedno več ljudi uporablja pametne mobilne naprave s čipom NFC. Pametne kartice so stalni spremljevalec pri vsakodnevnih opravilih. Uporabljajo se pri plačevanju na blagajnah in v večini primerov, ko ponudniki storitev izdajo plastično kartico. Vedno več jih podpira tudi brezstično komunikacijo NFC.

Cilj diplomskega dela je bil prikazati rešitev, ki nadomešča obstoječe brezstične pametne kartice z navideznimi brezstičnimi pametnimi karticami na pametni mobilni napravi NFC. Pri tem se je pazilo, da predlagana rešitev ni posegala v trenutno stanje. Rešitev omogoča vzporedno uporabo običajnih in navideznih kartic.

Opisan je pregled trenutnega stanja in možnosti, ki so razvijalcu na voljo. Trenutno najprimernejša rešitev je tista, ki uporablja varnostni element na MicroSD kartici, vsa komunikacija pa poteka preko centralnega strežnika. V prihodnosti se bo varnostni element najverjetneje prestavil v oblak. Opisane so obe rešitvi, s poudarkom na varnostnem elementu na MicroSD.

Ključne besede: NFC, brezstična pametna kartica, varnostni element, mobilna naprava, navidezna brezstična pametna kartica, JavaCard, MicroSD.

Abstract

Every day higher number of smart mobile devices with NFC chip are in use. Smart cards are already in use every day. They are used for payment and in most cases when companies issue plastic cards to their costumers. Number of those that are able of contact-less communication through NFC is rising.

Goal of thesis was to show the solution, that replaces existing contact-less smart cards with virtual contact-less smart cards on a mobile NFC devices. It was made sure, that solution didn't change the existing situation. Solution supports use of existing non virtual and virtual cards at the same time.

Current situation and possibilities that developer can use are described. Best solution at this moment is the one that uses secure element on MicroSD card and all the communication goes through central server. In future, the secure element will probably move to the cloud. Both solutions were described, focusing more on the one with MicroSD.

Keywords: NFC, contact-less smart card, secure element, mobile device, mobile phone, virtual contact-less smart card, JavaCard, MicroSD.

Poglavje 1

Uvod

Večina prodanih mobilnih telefonov spada med pametne telefone. Delež uporabnikov s pametnimi telefoni že predstavlja večino. Med pametnimi telefoni se dviguje tudi delež takšnih, ki podpira tehnologijo NFC.

Na trg množično prodirajo tudi pametne brezstične kartice. Veliko takšnih kartic se že uporablja. V Sloveniji in tudi v svetu se pojavljajo predvsem v transportnem prometu. Vedno pogosteje se pojavljajo tudi v obliki plačilnih kartic in kartic zvestobe. Sčasoma bodo brezstične pametne kartice zamenjale tudi stare kartice s čip in PIN, ki se uporabljajo predvsem pri bančnem prometu.

Ljudje imamo v denarnici veliko kartic. Prav tako imamo pri sebi vedno tudi mobilni telefon. Brezstične pametne kartice uporabljajo enako tehnologijo, kot se uporablja pri komunikaciji s čipi NFC na mobilnih telefonih. To pomeni, da lahko fizične pametne kartice simuliramo z navideznimi pametnimi karticami, ki jih pošiljamo preko oddajnika NFC na pametnem telefonu.

Ker bodo brezstične pametne kartice postale standard, to pomeni, da lahko denarnico preprosto pustimo doma, oziroma v njej hranimo le še fizičen denar pri gotovinskem plačevanju. Poleg tanjše denarnice pa uporabniku olajšamo tudi izdajo nove pametne kartice, saj je za to potrebnih samo nekaj klikov in za izdajo ni potrebno čakati v vrsti pred blagajno ali bančnim okencem. Bojazen, da bo uporabnik fizično kartico pozabil doma, s tem

odpade.

Prehod na digitalne navidezne kartice pa ne olajša poslovanja samo uporabniku, temveč tudi storitvenim podjetjem, ki takšne pametne kartice izdajajo. Izdaja fizičnih kartic tipično zahteva dolg in drag proces. Poleg same izdelave fizičnih kartic, mora storitveno podjetje poskrbeti za njihovo izdajo, polnjenje na polnilnih postajah ali na polnilnih okencih in skrbeti za izgubljene ter ukradene kartice. Podpora za navidezne kartice za storitveno podjetje zahteva malo ali nič dodatnih stroškov, saj se lahko uporabi obstoječo infrastrukturo bralnikov. Le da namesto brezstične pametne kartice uporabnik prisloni svojo mobilno napravo.

Cilj diplomskega dela je predstaviti model uporabe navideznih brezstičnih pametnih kartic, kakšno tehnologijo lahko uporabimo in kako poskrbimo za varnost prenosa podatkov ter tako preprečimo možne zlorabe. Na koncu bo predstavljen tudi predlog rešitve.

Poglavje 2

Predstavitev pojmov

2.1 Pametna kartica

Pametna kartica je plastična kartica velikosti in lastnosti, ki ustrezajo standardu ISO/IEC 7810 [1]. Vanjo je vgrajen sistem, ki je definiran po standardu ISO/IEC 7816 [2]. Vsebuje mikroprocesor, RAM, ROM, EEPROM in serijski vhod ter izhod preko kovinskega kontakta ali vgrajene antene. Kartice nimajo samostojnega napajanja, ampak se napajajo preko kontakta ali elektromagnetnega valovanja. Tega tipično dobijo od bralnika kartic ali terminala. Na njej se varno zapisujejo podatki. Pametne kartice se od navadnih pomnilniških razlikujejo v tem, da je vsebino pametne med uporabo mogoče spreminjati. Pri navadnih pomnilniških to ni mogoče, saj je vsebina nanjo zapisana med proizvodnjo. Pametne kartice zato poznajo nabor ukazov za dostop, branje in spreminjanje vsebine. Najbolj poznani primeri pametnih kartic so bančne kartice, osebne izkaznice, kartice zvestobe in kartice javnega prevoza.

2.1.1 Brezstična pametna kartica

Brezstična pametna kartica je pametna kartica, ki komunicira s terminalom ali bralnikom kartic preko radijskih valov. Definira jo standard ISO/IEC 14443 [3]. Komunikacija poteka po indukcijski tehnologiji, ki je podobna

RFID. Za vzpostavitev komunikacije je kartico potrebno približati na razdaljo nekaj cm od bralnika kartic, ki podpira brezstično komunikacijo. Transakcije kljub temu da ni fizičnega stika med kartico in bralnikom, potekajo hitro. S temi razlogi se pogosto uporabljajo tam, kjer čakalne vrste uporabnikov niso zaželene. Najpogostejše so kartice, ki se uporabljajo v transportu, zadnje čase pa izpodrivajo tudi obstoječe bančne kartice, ki komunicirajo z bralnikom preko fizičnega kontakta. Prve so izšle leta 2005 v ZDA, pojavljajo pa se tudi že v Evropi in Aziji. V Sloveniji se pričakuje uporabo brezstičnih pametnih kartic v letu 2014 [4]. Brezstični bralniki so v nekaterih trgovinah že prisotni.

2.1.2 Komunikacijski protokol

Pametna kartica deluje v razmerju nadrejeni/podrejeni (master/slave). Kartica je vedno v stanju pripravljenosti. Ko proti njej pride ukaz, na njega reagira z ustreznim odgovorom. Komunikacija poteka preko protokola APDU (application protocol data unit, slovensko podatkovna enota aplikacijskega protokola), definiran po ISO/IEC 7816-4 [2].

APDU predstavlja par ukaza in odgovora med bralnikom ter kartico. Ukaz in odgovor vedno nastopata v paru. Zato poznamo dva formata APDU-ja. Imenujemo jih kar ukaz APDU in odgovor APDU. Oba sta sestavljena iz niza bitov, večinoma grupiranih v bajte, včasih pa tudi v polbajte. Za lažje branje jih pišemo v šestnajstiški obliki (po dva znaka na bajt). Ukazni del para bralnik pošlje na pametno kartico, odgovor pa vrne pametna kartica nazaj bralniku.

Ukaz je sestavljen iz glave in telesa. Format ukaza je v Tabeli 2.1. V glavi imamo štiri bajte. Prvi je CLA (Instruction class), ki pove, za kakšen tip ukaza gre. Ta je odvisen predvsem od izvedbe in je tipično določen s strani podjetja, ki se te izvedbe loteva. Drugi bajt je INS (Instruction code), ki določa, za kakšen ukaz gre. Tipičen primer je ukaz "beri podatke". V tretjem in četrtem bajtu v glavi lahko podamo dodatne parametre, ki jih specifična izvedba pametne kartice potrebuje. V telesu imamo od 0 do 3 bajte Lc (Length command), s katerimi povemo, koliko bajtov podatkov

Glava (obvezna)				Telo (opcijsko)		
CLA	INS	P1	P2	Lc	Data field	Le

Tabela 2.1: ukaz APDU

Telo (opcijsko)	Rep (obvezen)	
Data field	SW1	SW2

Tabela 2.2: odgovor APDU

bomo poslali. Sledijo jim bajti za podatke v Data field. Število bajtov je definirano z Lc. Sledijo še od 0 do 3 bajti Le, ki pametni kartici sporočajo, kakšno dolžino odgovora od nje pričakujemo. 00(hex) na primer v večini izvedb pove, da pričakujemo vse podatke, ki so na voljo.

Odgovor je sestavljen iz telesa in repa. Format je prikazan v Tabeli 2.2. V opcijskem telesu so podatki, ki jih ukaz vrne. Pametna kartica ga tipično vrne pri branju podatkov. Dolžina je bila podana v ukazu. V repu pa sta dva statusna bajta, SW1 in SW2. Prvi je zopet odvisen od izvedbe in določen s strani proizvajalca, drugi bajt pa predstavlja status. Pove, ali je bil ukaz pravilno izvršen, če pa ni bil, pove, kaj je bil vzrok za neuspešnost. Odgovor 91 00(hex) tako na primer pomeni uspešno izvedbo.

2.1.3 Datotečna struktura

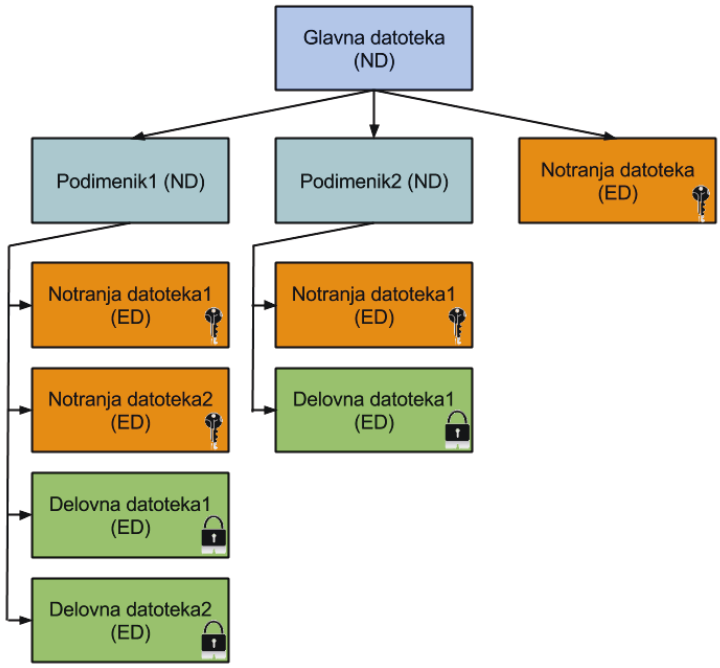
Datotečna struktura predstavlja razporeditev datotek na pametni kartici. Struktura, definirana po standardu ISO/IEC 7816 [5], pozna dva tipa datotek, namenske datoteke (dedicated file) in elementarne datoteke (elementary file). Namenske datoteke (ND) določajo hirearhijo, elementarne (ED) pa so namenjene shranjevanju podatkov. Slika 2.2 prikazuje primer takšne strukture.

Poznamo dve vrsti namenskih datotek. Namensko datoteko v korenu, ki jo izjemoma imenujemo glavna datoteka. Glavna datoteka je vedno priso-

tna in je ni mogoče brisati. Druge namenske datoteke je mogoče ustvariti in brisati. Imenujemo jih podimeniki in hirearhično predstavljajo nivo pod korenom. Vse namenske datoteke se uporabljajo za naslavljanje, nadzor in upravljanje elementarnih datotek. Namenske datoteke so na sliki predstavljene z modro barvo.

V elementarnih datotekah se shranjujejo ključi in podatki. Ključi se uporabljajo za šifriranje in dešifriranje podatkov. Elementarne datoteke se delijo na tiste, ki so namenjene shranjevanju ključev in tiste, ki so namenjene shranjevanju podatkov. Elementarne datoteke, ki so namenjene shranjevanju ključev, imenujemo notranje datoteke. Na sliki so prikazane z oranžno barvo. Na pametni kartici je vedno prisotna vsaj ena notranja datoteka. To je tista, ki jo hrani glavna datoteka. V njej se hrani glavni ključ (master key). Glavni ključ se uporablja za omejevanje urejanja podimenikov. Samo z njim lahko ustvarimo novi podimenik ali pa brišemo obstoječega. Ostale notranje datoteke so namenjene omejevanju dostopa do drugega tipa elementarnih datotek. To so delovne datoteke, na sliki obarvane zeleno. V njih se hranijo občutljivi podatki. Dostop je mogoč s ključi, ki so zapisani v notranjih datotekah.

Do glavne datoteke ali enega od podimenikov dostopamo tako, da jih naslovimo z AID naslovom. Naslavljanje se izvaja s posebnim ukazom APDU. Kartica vse ostale ukaze APDU namenja temu nivoju. Podimeniki si ne morejo deliti dostopa do notranjih in delovnih datotek. Te vedno pripadajo samo enemu od podimenikov. Samo ta podimenik lahko spreminja vsebino teh datotek. S tem je zagotovljena izoliranost podatkov. Glavna datoteka in pripadajoča notranja datoteka sta izjemi. Nista namenjeni shranjevanju in urejanju podatkov. Namenjena sta upravljanju z namenskimi datotekami. Ko je naslovljena glavna datoteka, se na kartici izvajajo ukazi APDU, namenjeni ustvarjanju, spreminjanju in brisanju podimenikov. Ko je naslovljen eden izmed podimenikov, pa se izvajajo ukazi APDU, namenjeni urejanju elementarnih datotek in njihove vsebine.



Slika 2.1: Datotečna struktura

2.1.4 Varnost

Spreminjanje vsebine ali pa branje kartice je določeno z varnostnimi statusi. Z njimi določimo, kateri načini pošiljanja ukazov so omogočeni. Poznamo več vrst statusov in več načinov pošiljanja [5].

Varnosti statusi

Prvi status je globalni varnostni status, ki določa, na kakšen način lahko dostopamo do glavne datoteke. V notranji datoteki glavne datoteke je shranjen glavni ključ (master key), ki ga ni mogoče izbrisati. Lahko se ga le spremeni. Ob izdelavi kartice je glavni ključ privzet. Menjavanje ključa je zaželjena praksa in v primeru privzetih vrednosti tudi nujna. Glavna datoteka je nekakšna vstopna točka in v trenutku, ko je kartica izdelana, tudi edina datoteka na kartici. Ostala struktura se med uporabo lahko spreminja, glavna datoteka pa nikoli. S tem statusom povemo, kako se lahko spreminja struktura, spreminja pa se lahko tudi varnostni status podimenikov.

Status podimenika določa, na kakšen način lahko dostopamo do delovnih datotek. Trenutna vrednost statusa je zapisana v notranjih datotekah, kjer se hranijo tudi ključi. Na tem nivoju lahko ključe, tako kot pri glavni datoteki, poljubno spreminjamo. Brisanje ključev je mogoče, vendar le na nivoju glavne datoteke.

Tretji status je status ukazov, je začasen in obstaja le med izvajanjem. Je tako imenovani delovni status in je odvisen od sosledja ukazov. V času izvajanja ukazov se preveri, ali se ujema z globalnim statusom ali pa statusom enega od podimenikov. Glede na primerjavo se omogoči, ali pa onemogoči uspešno izvedbo ukaza. Status ukazov se uporablja na vseh nivojih, za ukaze namenjenim glavni datoteki, podimenikom in ostalim notranjim ter delovnim datotekam.

Načini pošiljanja ukazov

Prvi način pošiljanja ukazov je pošiljanje v nešifrirani, prosti obliki. Uporabljali naj bi se pri prehajanju med glavno datoteko in podimeniki in pri branju prosto dostopnih podatkov. Statusi lahko omogočijo tudi prosto spreminjanje vsebine, strukture in branja občutljivih podatkov, vendar to iz očitnih razlogov ni priporočljivo. Tipičen dober primer uporabe takšnega načina, je pridobivanje enoličnega identifikatorja kartice, ki nam pove serijsko številko kartice.

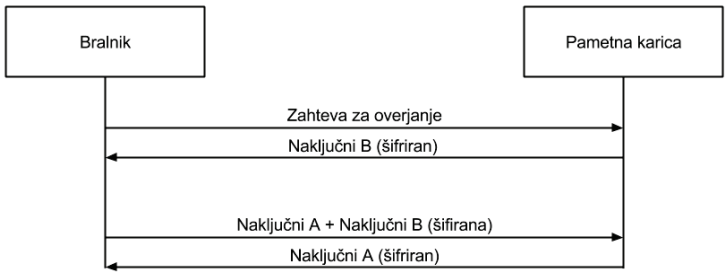
Drugi način pošiljanja ukazov, ki ga statusi lahko določajo, je pošiljanje s poznavanjem gesla. To je geslo, ki je tipično krajše oblike, tako imenovano osebna identifikacijska številka ali PIN (personal identification number). Ta način najpogosteje uporabljamo za zaščito uporabnikov, saj je geslo dovolj kratko, da si ga zapomnijo. V kombinaciji z omejitvijo napačnih poizkusov je dovolj, da kartice ne more uporabljati nekdo, ki ni hkrati tudi njen lastnik.

Tretji način pošiljanja ukazov je dokazovanje poznavanja ključa. To je daljše geslo, poznano izdajatelju kartice in pametni kartici. Ključ je zapisan v notranjih datotekah. Podatki na kartici se šifrirajo neposredno s ključem. Tak način ni najbolj varen, saj bi ob poznavanju vsebine morebitni napadalec ugotovil ključ. Uporaben je pri šifriranju naključnih podatkov, zato se ga uporablja pri overjanju oz. vzpostavitvi sejnega ključa.

Četrti način pa je šifriranje podatkov s sejnim ključem, določenim med overjanjem. Sejni ključ je začasen in primeren za šifriranje občutljivejših podatkov.

Overjanje

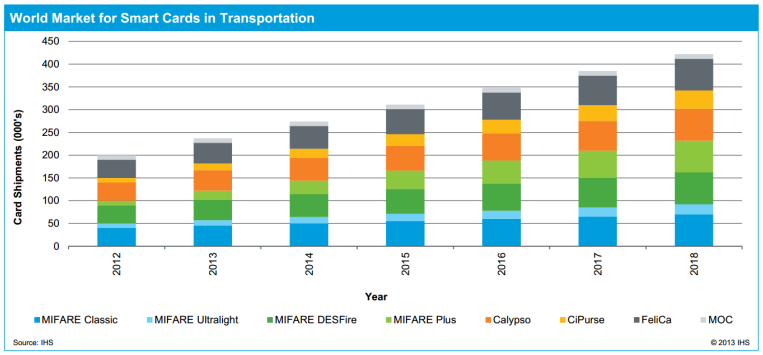
Overjanje je najpogostejši primer varnega prenosa podatkov. Z njim kartici dokažemo, da poznamo skriti ključ. Med overjanjem se zgradi sejni ključ, ki ga uporabljamo za šifriranje ukazov, ki mu sledijo. Na ta način zagotovimo, da potencialni napadalec ne more ugotoviti ključa, shranjenega v notranjih datotekah kartice. Med overjanjem, ko uporabljamo pravi ključ, šifriramo naključne podatke. Napadalec nima smiselnih podatkov, ki bi mu pomagali



Slika 2.2: Overjanje

pri ugotavljanju ključa. Kasneje, ko so podatki smiselni, pa se uporablja sejni ključ. Ključ je začasen, zato poznavanje ključa ob naslednji seji napadalcu ne koristi.

Slika 2.2 prikazuje tipičen primer overjanja. Bralnik kartici pošlje ukaz APDU, ki sproži začetek overjanja. Kartica odgovori z naključnim nizom bajtov B. Šifrira jih s ključem, zapisanim v notranjih datotekah. Ta je lahko v glavni datoteki ali pa v enem od podimenikov. Bralnik te nize prebere in dešifrira z istim ključem. Odšifriranim podatkom doda svoj naključni niz bajtov A. Oboje ponovno šifrira s ključem notranje datoteke. Združen in šifriran niz bajtov (A+B) proti kartici pošlje v drugem ukazu. Kartica niz zopet odšifrira. Dešifriran niz B primerja s poslanim v odgovoru prvega ukaza. Če kartica ugotovi, da se poslani in prejeti niz ujemata, potem ve, da pošiljatelj pozna ključ. V odgovoru drugega ukaza vrne šifriran niz A. Bralnik prejeti niz A odšifrira in primerja s poslanim. Če se bajti ujemajo s poslanimi, tudi on ve, da je naslavljal pravo kartico. S tem je overjanje zaključeno. V primeru, da je uspešno, bralnik in kartica iz teh dveh naključnih nizov znakov A in B sestavita po obema znanemu protokolu še sejni ključ. Kakšen je, je odvisno od implementacije. Tipično je sestavljen iz polovice niza A in polovice niza B.



Slika 2.3: Prodajni deleži kartic

2.2 Izvedbe brezstičnih pametnih kartic

Arhitekturo brezstičnih pametnih kartic določa standard ISO/IEC 14443. Kljub temu se pri fizičnih izvedbah podjetja, ki kartice izdelujejo, odločijo, da znotraj standarda uporabijo svoj način izvedbe. Standard namreč ne definira, kako mora biti logika na sami kartici izvedena. Lahko namreč dodajo svoje ukaze, ali pa definirajo več vrst delovnih datotek. Zaradi tega se lahko zgodi, da kartice različnih proizvajacev med sabo niso združljive. NFC Forum, ki skrbi za standard, se trudi prepričati proizvajalce, da bi popolnoma poenotili arhitekturo, vendar je proces združevanja počasen.

2.2.1 Komercialne izvedbe

Komercialne izvedbe pametnih kartic so najbolj razširjene. Napogosteje srečamo kartice Mifare (podjetja NXP Semiconductors), FeliCa (podjetja Sony) in Calypso (podjetje Innovatron). Na sliki 2.3 vidimo spreminjanje in napoved prodajnih deležev brezstičnih pametnih kartic v transportu[6].

Najbolj razširjene pametne kartice prihajajo iz podjetja NXP Semiconductors. Velik delež ima še vedno njihova brezstična pametna kartica Mifare Classic. Kljub temu, da je zelo razširjena, pa ni skladna s celotnim spektrom standarda ISO/IEC 14443. Še pomembnejše kot nestandardnost je to, da kartice tega tipa niso varne pred napadi. Uporabljaja Crypto-1 kriptirni

algoritem, ki pa je dokazano ranljiv[7].

Zaradi pomankljivosti je podjetje izdalo novejše različice, s katerimi so bile te pomankljivosti odpravljene. Classic so nadomestili z Mifare Plus, ki odpravi manjko varnosti, hkrati pa strukture ne spreminja drastično. Prilagodljivejša izvedba je kartica imenovana Mifare Desfire, ki jo NXP Semiconductors tudi najbolj priporoča. Kljub trudu, da bi ponudniki storitev, ki še uporabljajo Mifare Classic prešli na novejše različice, se ta prehod izvaja počasi. Tudi zato, ker novejšo različico, ki je skladna s standardi in posodobljeni varnostni algoritmi, spremljajo nove pogodbe. Te so za veliko podjetij drage, ob upoštevanju menjave vseh kartic uporabnikov pogosto predrage. Zaradi teh razlogov zaenkrat kljub neustreznosti ostajajo Classic kartice najbolj razširjene. Se pa delež počasi manjša.

Druge komercialne izvedbe kartic imajo manjši, vendar še vedno pomemben delež. MOC (match on card) so kartice, na katerih se nahaja biometrični prstni odtis v namene avtentikacije uporabnika. Zaradi varovanja osebnih poadtkov se pričakuje, da bo njihov delež kljub trenutni relativni popularnosti upadel.

2.2.2 Odprtokodne izvedbe

Komercialne izvedbe spremljajo drage pogodbe, saj morajo proizvodna podjetja plačevati licenčnino podjetjem, ki so jih načrtovala. Zato se je ustanovilo združenje OSPT Alliance. Združenje je pripravilo odprtokodno rešitev imenovano Cipurse. Za različne potrebe s razvili tri različne izvedbe. Cipurse L, Cipurse S in Cipurse T.

Cipurse L je najpreprostejša in najcenejša med njimi. Namenjena je enkratni uporabi, saj jo je zaradi cenejše izdelave po uporabi mogoče zavreči. Primer uporabe je dnevna avtobusna vozovnica.

Izvedba Cipurse S je namenjena večkratni, a še vedno preprosti uporabi. Primer uporabe je tedenska avtobusna vozovnica ali pa vozovnica z vnaprej znanim številom prevozov.

Izvedba Cipurse T pa je kompleksna, uporablja mikroprocesor, podpira

več hkratnih aplikacij in v popolnosti izkorišča vse možnosti, ki so dane s standardom ISO/IEC 14443. Proizvodnja je dražja, zato je namenjena uporabi v primerih, ko kartico potrebujemo dalj časa. Primer uporabe je letna avtobusna vozovnica s sliko. Zaradi kompleksne arhitekture pa je lahko poleg letne vozovnice uporabna tudi v druge namene. Lahko se uporablja kot vstopnica za kino, za plačevanje parkirnine, kot kartica zvestobe in podobno. Med popularnimi komercialnimi je podobna kartici Mifare DESfire, ki uporablja podobno arhitekturo in format ukazov.

OSPT Alliance ima vedno več podpornikov. Delež prodaje raste, zato se pričakuje, da bodo odprtokodne izvedbe v prihodnosti imele večjo vlogo. Cenejše licence pomenijo cenejšo proizvodnjo, pri tem pa ni ogrožena varnost, saj izvedba zahteva varne kriptirne algoritme.

2.3 NFC

NFC je nabor standardov za pametne mobilne naprave. Namenjen je vzpostavljanju brezžične povezave med napravami na zelo kratkih razdaljah. Tipično nekaj centimetrov. Uporablja se za identifikacijo, izmenjavo podatkov in vzpostavljanju kompleksnejših protokolov, kot sta bluetooth in wi-fi. Uporablja obstoječo tehnologijo RFID in standarde, določene v ISO/IEC 14443. Za nabor standardov skrbi organizacija NFC Forum, ustanovljena leta 2004, zadolžena je za razvoj, promocijo in certificiranje produktov, ki to tehnologijo želijo uporabljati.

V naslednjih poglavjih bodo predstavljeni ključni elementi ekosistema NFC. Ti elementi predstavljajo osnovo modela navideznih brezstičnih pametnih kartic. Model bo predstavljen v poglavjih, ki sledijo predstavitvi.

2.3.1 RFID

Radiofrekvenčno prepoznavanje ali RFID (radio frequency identification) je brezkontaktno pošiljanje ukazov preko elektromagnetnega polja. Prenos podatkov je hiter, vendar zaradi majhne pasovne širine počasen v primeru

pošiljanja večje količine podatkov.

RFID se uporablja na veliko področjih. Najpogosteje se uporablja značke RFID, ki imajo določene podatke zapečene na integrirano vezje. Omogoča lahko identificiranje značk brez kontakta na nekaj 10 metrov. Uporablja se pri brezkontaktnem plačevanju cestnine, vgrajene so lahko živalim pod kožo za lažjo identifikacijo, kot varnostne značke v trgovinah z oblačili in podobno.

Komunikacija NFC je nadgradnja RFID-a. Komunikacija poteka dvostransko in inicializator je lahko katerakoli naprava. Pri RFID je enostranska v smislu, da bralnik vedno začne komunikacijo z značkami.

2.3.2 Bralnik kartic

Bralnik je elektronska naprava, ki je namenjena branju kartic. Lahko ima samo vlogo vhodo-izhodne naprave v neki drug sistem, ki preko njega pošilja ukaze. Lahko pa je tudi samostojna enota in drugega sistema ne uporablja. Poznamo več vrst bralnikov. Prva vrsta so bralniki magnetnih zapisov na kartici, ki ob potegu skozi magnetno polje preberejo magnetni zapis. Druga so bralniki, ki preko posebnege reže preberejo zapis v čipu kartice. Tretja vrsta pa so bralniki, ki uporabljajo brezstično komunikacijo RFID.

Bralnike magnetnih zapisov ne srečujemo več pogosto, saj smo jih zaradi enosmerne komunikacije, pri kateri ne moremo zagotoviti varnost, zamenjali z bralniki čipa na kartici. Bralniki čipa podpirajo dvosmerno komunikacijo, napravi je priložena tudi tipkovnica, kjer je mogoče vnesti osebno številko PIN. Te naprave so v Sloveniji najpogostejše, saj jih vsakodnevno srečujemo na blagajnah trgovin in na bankomatih. Tretja vrsta bralnikov je trenutno manj razširjena, vendar počasi izpodriva drugo vrsto bralnikov. Obstajajo pa tudi hibridi, ki imajo dve ali pa celo vse tri vrste združene v eno napravo in se uporabljajo predvsem za mehkejši prehod iz ene tehnologije v drugo.

Brezstična komunikacija je hitrejša in tudi bolj higienična. Zato ni čudno, da so se začeli najprej pojavljati v javnem prometu, kjer sta ta dva atributa zelo pomembna. Pogosto so nameščeni tudi pred vrati poslovnih stavb. Preko njih odklepamo vrata, nekatera podjetja pa s tem sistemom preverjajo tudi

Slika 2.4: Na levi je bralnik ki podpira branje magnetnih zapisov in fizičnega čipa. Na desni bralnik, ki podpira vse tri načine branja.

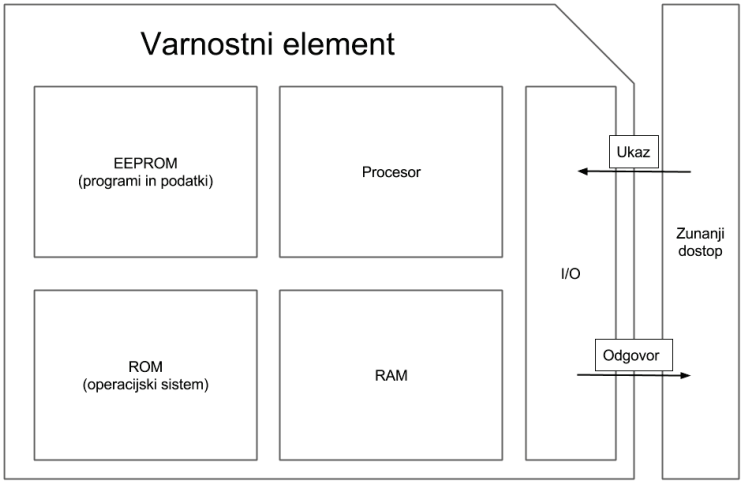
prisotnost zaposlenih na delovnem mestu.

2.3.3 Varnostni element

Varnostni element je integrirano vezje, ki zagotavlja visok nivo varnosti. V brezstičnem in tudi stičnem ekosistemu je namenjeno varnemu shranjevanju podatkov. Njegova ključna lastnost je, da ni mogoč direkten dostop do pomnilnika. Do njega dostopamo z ukazi, ki zahtevajo poznavanje varnostnega protokola in gesla.

Čip ima svoj procesor, RAM, EEPROM in ROM. Operacijski sistem je shranjen v ROM, v EEPROM pa shranjujemo programe in podatke. Struktura je prikazana na Sliki 2.5 Podpira več različnih varnostnih protokolov, kot so DES, 3DES, RSA in druge. Dostop je močno omejen in mogoč le ob poznavanju ključev preko vhodno-izhodnih ukazov. Programi, ki tečejo na varnostnem elementu, se zato izvajajo v zaprtem okolju.

Varnostni element v takšni ali drugačni obliki srečamo v vseh pametnih karticah. Dostop do podatkov na kartici je najpogosteje možen le izdajatelju kartice, ne pa tudi imetniku kartice. Uporabnik tako ne more spreminjati vsebine zapisa. S tem mu onemogočimo potvarjanje podatkov. Hkrati preprečimo tudi ponarejanje potencialnih napadalcev. Pred krajo varnostnega elementa lahko uporabnika zaščitimo s številko PIN, ki je znana samo njemu.



Slika 2.5: Struktura varnostnega elementa

Vse pametne kartice vsebujejo varnostni element.

2.3.4 OTA

Mobilne naprave za prenos podatkov uporabljajo brezžične tehnologije. Te so precej dovzetnejše za napade, saj je napadalcu za vdor dovolj že to, da se nahaja v relativni bližini.

Da bi lahko zagotovili varen prenos med napravami, moramo poskrbeti, da so podatki med prenosom kriptirani in znani samo napravama, ki si zaupata. Za to komunikacijo vzpostavimo tako imenovani varni kanal preko zraka ali OTA (over the air secure channel). Shranjevanje v pomnilniku mobilne naprave ni varno, saj je prosto dostopno. Prav tako ni zaščiten dostop do procesorja v mobilni napravi. Edino mesto, kjer so podatki lahko zaščiteni, je v varnostnem elementu samem.

Zato mora biti kanal OTA vzpostavljen med varnostnim elementom in napravo, ki ima dovoljenje za dostop do podatkov. Mobilna naprava je v tem primeru zgolj posrednik. Ker ključev ne pozna, ji vsebina ni znana.

2.3.5 MNO

Mobile network operator ali MNO (mobilni operater) je ponudnik mobilnih storitev. Končnim uporabnikom ponuja mobilno telefonijo, prenos podatkov, skrbi za nemoteno delovanje omrežja, prodajo, marketing in podobno.

MNO svojim uporabnikom izda kartico SIM. Z njo uporabniki dostopajo do mobilnega omrežja. Kartica SIM je pametna kartica in vsebuje varnostni element. V njej so zapisani podatki o telefonski številki. Del vsebine je prosto dostopen in ga lahko uporabnik uporablja za shranjevanje telefonskih številk in zgodovino klicev.

2.3.6 Ponudniki storitev

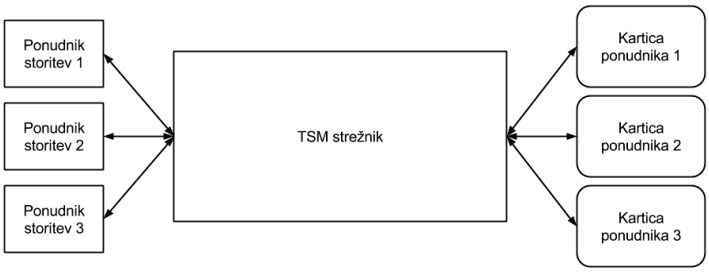
Ponudnik storitev je podjetje, ki svojim strankam nudi storitve. Trgovska veriga, transportno podjetje, banke in drugi spadajo mednje. Da bi podjetja svojim strankam olajšala poslovanje, jim izdajo pametne kartice. To so najpogosteje kartice zvestobe, plačilne kartice in mesečne ter druge vozovnice.

2.3.7 TSM

Trusted service manager (zaupanja vreden upravljavec storitev) je vloga v ekosistemu NFC, ki ima vlogo posrednika med mobilnimi operaterji, proizvajalci mobilnih naprav, izdajatelji kartic, uporabniki in varnostnim elementom.

Njegova vloga je, da zagotavlja varnost v tem ekosistemu. V njem ima najodgvornejše delo. Poskrbeti mora, da se avtorizira uporabnike, avtenticira kartice in prepreči potencialne napade. Nuditi mora neprestano podporo in njihovi strežniki morajo biti neprestano dosegljivi.

TSM administrira vse aplikacije, ki prihajajo od različnih ponudnikov storitev. Preverja avtentičnost njih in njihovih aplikacij in jim dovoljuje oziroma omejuje dostop do strežnika TSM in prostora na varnostnem elementu, ki ga TSM upravlja.



Slika 2.6: TSM posredništvo

Vloga TSM je ločena od ponudnika storitev, vendar lahko obe vlogi zaseda isto podjetje. Za združitev vlog se po navadi odločajo večja podjeta, ki imajo dovolj strank, da si lahko privoščijo imeti to vlogo. Manjša podjetja pa vlogo TSM po navadi prepustijo zunanjim izvajalcem.

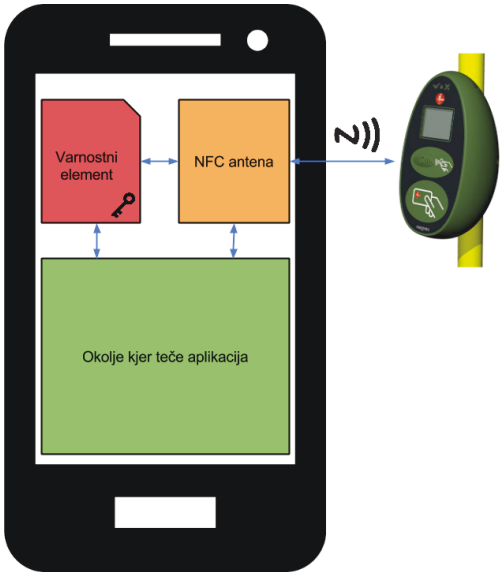
Povezava vlog je prikazana na sliki 2.6.

2.3.8 Povezanost komponent NFC

Na sliki 2.7 vidimo, kako se deli arhitekture NFC povezujejo. Da lahko govorimo o okolju NFC, potrebujemo vsaj tri komponente: anteno NFC, varnostni element na mobilni napravi in zunanji bralnik. Antena NFC v tem primeru deluje kot posrednik med varnostnim elementom in bralnikom.

Do varnostnega elementa in antene na mobilni napravi lahko dostopamo tudi aplikativno. Z aplikacijo, ki teče na mobilni napravi, lahko pošiljamo enake ukaze kot zunanji bralnik. Preko nje lahko dostopamo neposredno ali pa preko bralnika.

Prikazana arhitektura je splošna in vse vrste komunikacij niso nujno podprte. Raznolikost izvedb privede do tega, da nekateri komunikacijski kanali niso podprti. V nekaterih primerih aplikacija nima neposrednega dostopa do varnostnega elementa in lahko do njega dostopa le preko NFC. Pogosto sta varnostni element in antena NFC na istem fizičnem čipu, ni pa to nujno.



Slika 2.7: Splošna NFC arhitektura

2.4 JavaCard

JavaCard je tehnologija programske opreme, ki omogoča, da aplikacije, pisane v Javi, tečejo na pametnih karticah [8]. Tehnologija je v skladu z ISO/IEC 7816-4 standardom. S tem razlogom je veliko implementacij varnostnih elementov pisanih v JavaCard tehnologiji. Programi, ki tečejo na njej, se imenujejo appleti. Na JavaCard karticah je implementiran okrnjen JVM, ki izvaja applete, ki so prevedeni z Javanskim interpreterjem. Kartica svojega napajanja nima, zato je tako kot druge pametne kartice odvisna od zunanjega. Kartica izvaja ukaze v začasnem pomnilniku in jih v primeru uspešne izvedbe shrani na trajnega. S tem je zagotovljeno vedno veljavno stanje v primeru, da kartico v napačnem trenutku odmaknemo od napajanja.

Na kartici je lahko naloženih več programov vzporedno, vendar je lahko le eden izmed njih trenutno izbran in aktiven. Za to poskrbi JVM, ki določa, komu so ukazi namenjeni. Programe imenujemo mobilni appleti. Vsak izmed njih ima svojo prostorsko domeno. Appleti so ločeni v času izvajanja in ne morejo dostopati do podatkov drugih appletov. Za to skrbi posebni

požarni zid. JavaCard podpira več vrst kriptirnih algoritmov. Podpira najpogostejše s simetričnimi ključi, to so DES, 3DES in AES. Podpira pa tudi nesimetrična RSA in kriptografijo eliptičnih krivulj. Dostop do JavaCard je mogoč samo preko ukazov APDU. Z njimi na kartico nalagamo aplete, jih brišemo, urejamo in izbiramo. Prav tako pa z njimi pošiljamo tudi ukaze za izbran applet. Naslov appleta je po standardu določen z naslovom AID.

Razlika med njimi in drugimi bolj razširjenimi oblikami pametnih kartic je ta, da lahko na njej teče več kot en program. Na bolj razširjenih karticah je izvajanje zapečeno na kartico. Ker je izvedeno v obliki fizičnega vezja, reprogramiranje ni možno. JavaCard reprogramiranje omogoča, saj lahko sami napišemo poljuben program in ga naložimo nanjo. Vsak od njih lahko simulira ali implementira različne izvedbe pametnih kartic. Zaradi prilagodljivosti in programibilnosti pa je izvajanje na JavaCard počasnejše od standardnih pametnih kartic. Predvsem zato, ker so bili šifrirni algoritmi razviti za namene strojnih in ne programskih rešitev.

2.4.1 Mobilni applet

JVM na JavaCard je podmnožica Java Standard Edition. Na JavaCard zato ne poznamo arhetipov char, int, long, double in float. Prav tako manjkajo kvalifikatorji, kot so enum in transient. Polja so lahko samo enodimenzionalna, manjka kloniranje objektov, delo z nitmi ni omogočeno. Manjka tudi sproščanje pomnilnika in mora za to poskrbeti razvijalec sam. Pomanjkanje prostora na kartici močno okrni tudi nabor knjižnic, ki se uporabljajo v Java SE.

Leta 2008 je izšel JavaCard 3.0 z razširjenim JVM, ki je dodal mrežno povezljivost. Mogoče je pisati servlete s podporo HTML, REST in SOAP. Dodano je tudi sproščanje pomnilnika in več nitnost. Žal pa so strojne rešitve, ki bi podpirale novo različico, še zelo redke.

Zaradi podobnosti s standardno Javo je krivulja učenja za poznajalce Jave precej hitrejša kot pri drugih oblikah pametnih kartic. Prav tako je uporabnih precej algoritmov, ki z nič ali malo sprememb tečejo tako na JavaSE kot na

JavaCard.

2.5 Ekosistem brezstičnih pametnih kartic

Ekosistem brezstičnih pametnih kartic je povezanost komponent NFC. Način povezanosti določa, na kakšen način je mogoče uporabljati pametne kartice. Ta način povezanosti imenujemo model.

Za lažjo razlago modelov bodo za primer brezstičnih pametnih kartic uporabljene kartice, ki jih uporabljajo javna prevozna podjetja. Ta nastopajo kot ponudnik storitev. Model je podoben ali enak tudi pri drugih oblikah uporabe, vendar so bralcu brezstične kartice s področja transporta najbolj poznane. Kartico lahko podjetje, ki izvaja prevoz potnikov, uporablja na različne načine. Na kartici so lahko zapisani podatki o tem, koliko voženj ima uporabnik na razpolago ali pa je na njej zapisano trenutno stanje sredstev, ki jih uporabnik lahko še porabi. Lahko je uporabljena tudi kot tedenska, mesečna ali letna vozovnica. Takrat so na njej zapisani podatki o veljavnosti, lastniku in tipu vozovnice.

Uporabniki pri uporabi kartice prislonijo kartico k bralniku. Ti bralniki so lahko postavljeni na vhodu v postajo, na vlaku ali avtobusu in povsod tam, kjer ponudnik prevozov želi preveriti ali spremeniti stanje na kartici. Prav tako mora uporabnik kartico prisloniti na posebne bralnike, imenovane polnilne postaje. To so večje naprave, kjer uporabnik dokupi vozovnice, poveča stanje, podaljša veljavnost vozovnice in podobno. Zato imajo te naprave posebne reže za plačilne kartice ali gotovino. Polnilno postajo lahko nadzoruje tudi blagajnik. V tem primeru reže za kartico in gotovino ni. Za pisanje vrednosti na kartico v tem primeru poskrbi blagajnik.

2.5.1 Obstoječi model

Obstoječi model, ki se uporablja pri plačevanju voženj z brezstičnimi karticami, je sestavljen iz treh elementov: plačilne pametne kartice, bralnikov in strežnika, s katerim nekateri bralniki komunicirajo. Model je prikazan na

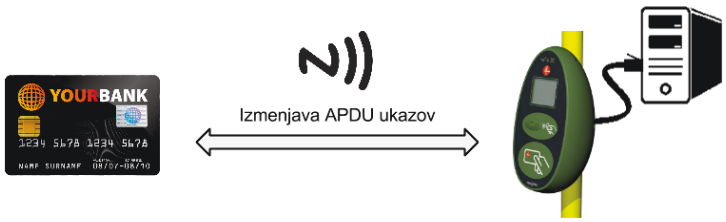
Sliki 2.8.

Pametna kartica je personalizirana in pripravljena za uporabo že v času izdaje. Personalizacija je postopek, s katerim na kartico naložimo datotečno strukturo, ključe in začetne vrednosti. To uporabnik tipično kupi na blagajnah v bližini postaje. Na istih blagajnah lahko uporabniki tudi podaljša ali dokupi vozovnice. Ti bralniki so povezani s strežnikom predvsem zaradi podatkov, potrebnih pri plačevanju. Ko želi uporabnik dokupiti vožnje, približa kartico bralniku. Ta medtem ves čas čaka na kontakt. Ko zazna kartico, sproži niz ukazov APDU in čaka na odgovore. Bralnik prejete odgovore preko povezave posreduje strežniku. Ta ponavadi preveri stanje na plačilnih računih, veljavnost dovoljenj in podobno.

Druge vrste bralniki so tisti na vhodih postaj, avtobusov in vlakov. Tam bralniki ne komunicirajo s strežnikom. Polnjenje in podaljšanje veljavnosti vozovnic na njih ni mogoče. Ti bralniki lahko na kartice zapisujejo vhodno in izhodno postajo, odštejejo sredstva na njej ali pa samo preverijo veljavnost. Komunikacija z oddaljenim strežnikom v tem primeru ni potrebna, saj ima bralnik omogočen dostop do segmentov na kartici.

Obe vrsti bralnikov se na videz in primer uporabe bralnika razlikujeta. Vendar je način delovanja za oba enak. Razlika je le v tem, ali bralnik pošilja podatke na neki oddaljen strežnik ali pa je analiza podatkov izvedena lokalno. Pri prvem načinu se ključi lahko hranijo na bralnikih ali strežniku, v drugem primeru pa se uporabljajo samo na bralniku.

Pogosto srečamo tudi bralnike, ki obdelujejo podatke lokalno, se pa občasno povezujejo na oddaljen strežnik. To se pri potniškem prometu uporablja za tako imenovane kartice na črni listi. Nekateri prevozniki se zaradi zlorab ali vandalizma odločajo za blokiranje uporabe nekaterih kartic. Zato bralniki, ki delujejo lokalno pri preverjanju, uporabljajo shranjen seznam takšnih kartic. Seznam se mora v tem primeru občasno posodobiti. To se ponavadi zgodi enkrat dnevno, na posebnih postajah, kjer je dostop do oddaljenega strežnika mogoč. V tem primeru ni potrebe po hitri sinhorinizaciji seznamov.

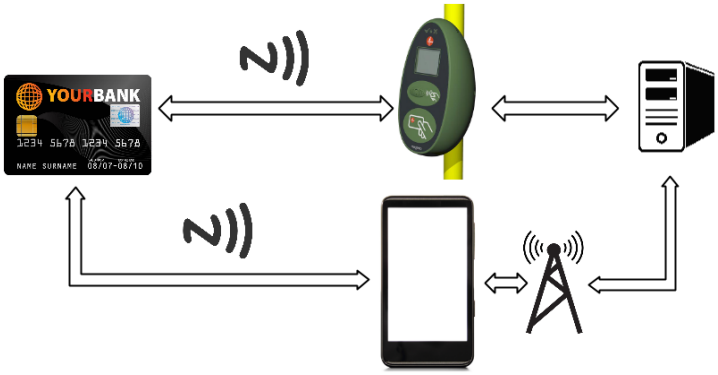


Slika 2.8: Obstoječi model

2.5.2 Vpeljava mobilne naprave

Obstoječi model je tisti, ki ga najpogosteje srečujemo. Lahko pa takšen model nadgradimo z manjšo spremembo, ki ne zahteva veliko vlaganja. V model vpeljemo mobilno napravo, ki podpira komunikacijo NFC. Nadgrajen model je prikazan na Sliki 2.9. Novi model ohrani obstoječi model s fizično pametno kartico. Še vedno uporablja pametno kartico, bralnik in strežnik. Dodatek je mobilna naprava, ki se v sistem povezuje kot dodaten bralnik. Prav tako kot bralniki v obstoječem sistemu se ukazi APDU pošiljajo preko oddajnika NFC na mobilni napravi. Mobilna naprava se preko mobilnega internetnega omrežja povezuje na oddaljen strežnik. V tem primeru mobilna naprava uporablja enake protokole kot bralnik, saj le nadomešča nekatere bralnike.

Branje kartic z mobilno napravo omogoča uporabniku, da podatke na kartici nadgrajujejo ali berejo sami, s svojo pametno mobilno napravo. Če želijo podaljšati veljavnost mesečne vozovnice ali pa dodati sredstva na kartico, jim ni treba čakati v vrstah pred blagajno. Uporabnik lahko prav tako sam s svojo mobilno napravo preveri trenutno stanje na kartici. Da bi lahko uporabnik izvedel nadgradnjo stanja na svoji brezstični pametni kartici, potrebuje dvoje. Pametno mobilno napravo s podporo NFC in pa posebno aplikacijo, ki posreduje podatke med uporabnikom in oddaljenim strežnikom. V tem primeru mobilna naprava deluje le kot posrednik in ni sposobna sama brati in spreminjati podatkov. Zato mora mobilna naprava v času branja in pisanja imeti povezavo z internetnim omrežjem. Tako lahko zagotovi, da strežnik



Slika 2.9: Mobilna naprav kot bralnik

šifrira podatke in ukaze, ki jih pametna kartica dešifrira. Pametna kartica v odgovoru preko mobilna naprave pošlje nazaj proti strežniku.

Podjetje, ki izvaja prevoze, lahko ob razširjeni uporabi mobilne naprave kot bralnika zmanjša število bralnikov, ki jih sicer uporablja. Prav tako lahko prevozna podjetja uporabljajo mobilno napravo kot bralnik na avtobusnih postajah, avtobusih ali vlakih.

2.5.3 Naš model

Izraz naš model uporabljamo, ker gre za model, razvit v našem podjetju. Prva dva modela uporabljata fizično brezstično pametno kartico. Naš model pa poleg fizične kartice, vpelje navidezno brezstično pametno kartico. Sistema s fizično in navidezno kartico uporabljata enake bralnike in obstoječo infrastrukturo. Razlika je le ta, da pri navidezni pametni kartici ne potrebujemo fizične pametne kartice. Navidezna je zato, ker jo simuliramo na mobilni napravi in obstaja le v digitalni obliki.

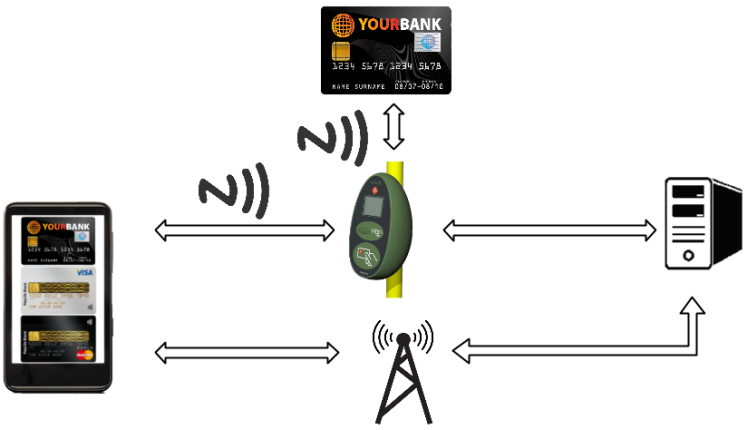
Fizična brezstična pametna kartica uporablja varnostni element na kartici. Ta prepreči morebitne vdore in nepooblaščeno spreminjanje vsebine kartice. Pri navidezni pametni kartici želimo imeti enak nivo varnosti. Zato tudi tu potrebujemo varnostni element, le da ta ni v obliki čipa na fizični

kartici. V naslednjem poglavju bodo predstavljeni tipi varnostnih elementov skupaj s prednostmi in slabostmi.

Uporabnik, ki želi potovati z javnim prevoznim sredstvom, v novem modelu za plačilo in identifikacijo vozovnice uporabi kar svojo pametno mobilno napravo s podporo NFC. Za to ne potrebuje fizične pametne kartice. Naložiti mora posebno aplikacijo in jo povezati z izbranim varnostnim elementom. Ko to stori, lahko kupuje vozovnice več različnih ponudnikov storitev. Brez dodatne nadgradnje lahko uporablja navidezne pametne kartice različnih ponudnikov storitev. Ne samo v javnem prevozu, temveč tudi za druge storitve, kot so kartice zvestobe, plačilne kartice, kartice namenjene odklepanju prostorov in parkrišč. Povsod tam, kjer se že ali pa se še bodo uporabljale brezstične pametne kartice. Ko uporabnik želi kupiti novo karto ali pa podaljšati veljavnost obstoječe, odpre mobilno aplikacijo. Za nakup potrebuje internetno omrežje, preko katerega prenese ukaze APDU in izvede plačilo. Po tem postopku uporabnik uporablja svojo mobilno pametno napravo namesto fizične brezstične kartice na povsem enak način. Približa jo bralniku in plača ali pa validira vozovnico.

Mobilna naprava skupaj z ustreznim varnostnim elementom za razliko od fizičnih pametnih kartic omogoča hranjenje več navidezih kartic hkrati. Omejena je le s prostorom na varnostnem elementu. Brez težav lahko uporablja nekaj 10 navideznih pametnih kartic tudi ob najbolj prostorsko omejenem varnostnem elementu. V vsakem trenutku lahko zamenja vsebino starih kartic z novimi. Za izdajo nove navidezne kartice uporabnik potrebuje le nekaj klikov. Brez čakanja v vrstah, izpolnjevanja novih obrazcev in plačila, ki pogosto spremlja novo fizično brezstično kartico.

V zadnjem poglavju bo predstavljena rešitev našega modela. Ta omogoča varno hranjenje in uporabo navideznih brezstičnih pametnih kartic na mobilni napravi. Pri tem nam obstoječega sistema večinoma ni potrebno spreminjati. Uvesti moramo samo dodatno storitev, ki omogoča vzporedno uporabo.



Slika 2.10: Naš model

Poglavje 3

Tipi varnostnih elementov

3.1 Kartica SIM

Kartica SIM (krajše za subscriber identification module) je integrirano vezje na kartici manjših dimenzij. Uporabljajo jo vsi mobilni operaterji za varno shranjevanje podatkov o njihovem naročniku. Je nepogrešljiv element mobilnih telefonov in drugih naprav, ki uporabljajo mobilno omrežje. Vezje SIM je vgrajeno v plastično kartico in zato prenosljivo med mobilnimi napravami.

Vsaka kartica SIM je enolično določena s serijsko številko ICCID. Ta je zapečena na vezje med proizvodnjo ali pa nepovratno zapisana med personalizacijo. V ICCID so med drugim zapisani podatki o proizvajalcu čipa, državi in izdajatelju kartice. ICCID določa uporabnika in omogoča mobilnemu operaterju posredovanje podatkov med napravami v njihovem ali tujem mobilnem omrežju. Najpogosteje posreduje klice in tekstovna sporočila.

Kartica SIM je varnostni element, saj vsebuje datotečno strukturo in ključe, ki omogočajo varno posredovanje podatkov. Ključ je zapisan v kartico in informacijo o njem poznata samo kartica ter mobilni operater. Uporabnik sam podatkov na njej ne more spreminjati, ker nima informacije o njem. Zato lahko zapisovanje podatkov na kartico SIM ali iz nje omogoča samo MNO, oziroma podjetje, ki ima vlogo TSM v primeru, da jo MNO ne opravlja sam.

3.1.1 Prednosti

Največja prednost kartice SIM v vlogi varnostnega elementa je ta, da jo imajo vsi mobilni telefoni. Zaradi identifikacije in pozicioniranja uporabnika je nepogrešljiv element. Druga velika prednost je ta, da kartice SIM pogosto uporabljajo JavaCard tehnologijo in so programibilne. Novejše različice kartic vgrajeno tudi anteno NFC, kar je prednost takrat, ko take antene mobilna naprava nima. Prav tako na novejših različicah najdemo aplete, ki podpirajo določene izvedbe navideznih pametnih kartic.

3.1.2 Slabosti

Kartica SIM je v lastništvu mobilnih operaterjev. Če želimo uporabiti SIM za varnostni element, moramo za dostop pridobiti dovoljenje. Mobilnih operaterjev je veliko. To pomeni veliko različnih sistemov, saj mobilni operaterji uporabljajo različne tipe kartic SIM. Nekatere kartice SIM podpirajo programibilnost in nalaganje appletov, druge ne. Operaterji, tipično za vlogo TSM, najamejo drugo podjetje. V tem primeru moramo za dostop do kartice SIM poleg MNO zaprositi tudi podjetje TSM. Takšna dovoljenja je po navadi zelo težko dobiti.

Nepodprtost programibilnosti kartic SIM in veliko podjetij, od katerih moramo dobiti dostop, pomeni zmanjšan nabor mobilnih naprav, ki jih naš sistem lahko podpre. To je velika ovira pri izboru kartice SIM za varnostni element.

3.1.3 Primer

Primer kartice SIM najdemo v vsakem mobilnem telefonu, pa tudi v nekaterih drugih mobilnih napravah. Uporabnik jo pridobi od mobilnega operaterja. Primeri kartic SIM so prikazani na sliki 3.1.

Slika 3.1: Primeri SIM kartic

3.2 Vdelani varnostni element

Vdelani varnostni element je varnostni element, vdelan v mobilno napravo. Vdela ga proizvajalec med proizvodnjo. Tipično je to v obliki ločenega čipa z omejeno dostopnostjo. Glavni ključ je določen med proizvodnjo. Najpogosteje proizvajalci vgrajene varnostne elemente dodajo v pametne telefone za lastne potrebe. Varnostni element uporabijo za nekatere aplikacije, ki visok nivo varnosti potrebujejo. Zadnja leta se nekateri proizvajalci odločajo, da del varnostnega elementa ponudijo tudi drugim razvijalcem. Realizacija dostopa ni standardna in je odvisna od proizvajalca.

3.2.1 Prednosti

Proizvajalcem, ki dostop do varnostnega elementa dovoljujejo, je v interesu, da uporabniki varnostni element uporabijo. Zato je je dostop do varnostega elementa razmeroma preprost. Dovolj je že, da zanj zaprosimo. Ko to dovoljenje imamo, je dostop enostaven. Proizvajalec ima v takih primerih že na-

pisan API preko katerega dostopamo do svojega dela varnostnega elementa. Prav tako poskrbi za varnost dostopa do varnostnega elementa in varnost na njem samem.

3.2.2 Slabosti

Prva slabost je ta, da so mobilne naprave, ki vsebujejo varnostni element, še zelo redke in tipično prisotne le v telefonih višjega cenovnega razreda. Drugi problem je, da je dostop do varnostnega elementa na mobilni napravi za veliko naprav onemogočen in ga proizvajalec uporablja v lastne namene. S temi razlogi izbira vdelanih varnostnih elementov pomeni izgubo velikega dela potencialnih uporabnikov. Delež mobilnih naprav z vgrajenimi in dostopnimi varnostnimi elementi je majhen. Poleg tega moramo za vsakega proizvajalca ali pa celo za posamezne modele mobilnih naprav prilagoditi naš sistem. Kot rečeno, proizvajalci nimajo skupnega načina implementacije. Z dostopom do varnostnega elementa na enem od modelov ali znamk si pri drugih ne moremo pomagati.

Varnostni element je v tem primeru vezan na mobilno napravo in ga ni mogoče prenašati med napravami. Prenašamo lahko vsebino na njem, vendar to zahteva dodatno delo.

3.2.3 Primer

Vdelan element v nekaterih telefonih višjega ranga uporablja proizvajalec mobilnih naprav Samsung. Ta je v svoje novejše naprave začel vgrajevati takoimenovani Knox sistem, ki razvijalcem omogoča, da njihove aplikacije dobijo poseben prostor na varnostnem elementu, vgrajenem v drobovju vezja. Do njega lahko dostopa samo odgovorni za varovanje podatkov te aplikacije. Dodeljen jim je tako imenovani vsebnik (container), dostop do njega pa določen s ključem AES . Znotraj njega lahko razvijalec hrani občutljive podatke. Tudi Samsung uporablja Knox. V njem ima zapisane podatke o tem, ali je bil na telefonu izveden korenski (root) dostop. Uporablja ga tudi za

druge svoje varne aplikacije. Knox sistem je v prvi vrsti namenjen poslovnim uporabnikom, ki želijo nezaželenim osebam preprečiti dostop do varnostno občutljivih podatkov.

3.3 MicroSD

SD (Secure digital) je spominska kartica, namenjena shranjevanju podatkov. Uporablja bliskovni (flash) pomnilnik. Je majhna in prenosljiva, zato se jo pogosto uporablja pri manjših napravah, kot so fotografski aparati, naprave GPS, prenosniki in druge prenosljive naprave. MicroSD je približno štirikrat manjši od navadega SD in je velikosti nohta na prstu.

MicroSD sam po sebi ni varnostni element. Da to postane, mora biti v njega vgrajen poseben čip, ki lahko zagotavlja potrebno varnost.

Varnostnemu elementu v MicroSD obliki način dostopa določa proizvajalec oziroma prodajalec. V času prodaje kupcu dodelijo dostop. Dostop do varnostnega elementa je mogoč s privzetimi ali pa priloženimi ključi. Novi lastnik nato nanj naloži svoje podatke, kluče ali programe. Programe lahko nalaga v primeru, da varnostni element podpira JavaCard ali kako drugo podobno tehnologijo.

3.3.1 Prednosti

Velika prednost varnostnega elementa je ta, da je v popolnoma prilagodljiv. Lasntnik MicroSD je popolnoma neodvisen od vlog TSM in MNO. Ključe dobi ob prodaji, ne da bi za ta dostop moral posebaj zaprositi. Veliko kartic MicroSD podpira JavaCard tehnologijo. To pomeni, da je kartica programibilna in da nanjo lahko nalagamo poljubne applete.

3.3.2 Slabosti

Prva slabost MicroSD je ta, da mora mobilna naprava vsebovati posebno režo, v katero lahko vstavimo varnostni element. Veliko naprav takšne reže

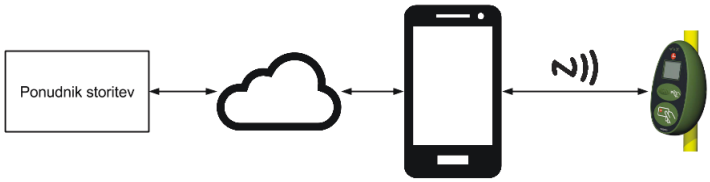
Slika 3.2: kartica MicroSD podjetja DeviceFidelity

nima. Prav tako se pojavi težava takrat, ko uporabnik to režo že uporablja kot razširitveno pomnilno mesto z neko drugo kartico MicroSD, ki pa ni pod našim nadzorom, ali pa na njej ni varnostnega elementa.

Druga slabost pa je ta, da je MicroSD tako imenovana dodatna oprema. To pomeni, da mobilne naprave ob nakupu nimajo priloženih ustreznih kartic MicroSD in jo je za to potrebno naknadno kupiti. To pomeni dodaten strošek za uporabnika ali pa implementatorja našega modela.

3.3.3 Primer

Primer take uporabe je MicroSD podjetja DeviceFidelity, ki vanj vgrajuje varnostni element. Poleg varnostnega elementa je v kartico vgrajena antena, ki omogoča komunikacijo NFC. Njihov MicroSD je implementacija JavaCard-a. To naredi napravo programibilno, saj lahko razvijalci napišejo applet, ki implementira tip varnostnega elementa po svoje.



Slika 3.3: Varnostni element v oblaku

3.4 Oblak

Varnostni element je tipično v obliki fizičnega vezja in se nahaja nekje v bližini naprave (SIM, vgrajen element, MicroSD), vendar to ni pogoj. Varnostni element se lahko nahaja tudi na oddaljenem strežniku ali oblaku. Implementacija na strežniku je enaka tisti v oblaku. Oblak le poskrbi za skalabilnost in hitrost dostopa.

Za uporabo varnostnega elementa v oblaku potrebujemo dvoje. Prvo je potrebno implementirati varnostni element na strežniku. Pri drugih oblikah varnostnih elementov, ki so implementirani fizično, je že poskrbljeno za varnost, podporo ukazom APDU in izolacijo podatkov. Prav tako so vanje že vgrajeni šifrirni čipi. V oblaku tega nimamo in moramo poskrbeti, da podpremo vse funkcionalnosti fizičnih implementacij. Poskrbeti je potrebno tudi za enak nivo varnosti.

Poleg implementacije varnostnega elementa je potrebno poskrbeti tudi za prenos podatkov med mobilno napravo in strežnikom. Za to naprava potrebuje imeti internetno povezavo do strežnika.

3.4.1 Prednosti

Velika prednost implementacije varnostnega elementa v oblaku je, da je nadzor nad njem prepuščen razvijalcu. Je povsem neodvisen od vlog TSM in MNO, proizvajalcev mobilnih naprav in proizvajalcev varnostnih elementov. Prav tako je na strežniku lažje implementirati različne izvedbe varnostnih

elementov. Programiranje strežniške kode je lažje od pisanja appletov v JavaCard tehnologiji.

3.4.2 Slabosti

Glavna omejitev varnostnega elementa v oblaku je odvisnost od mobilnega internetnega omrežja. Naš model ni mogoče uporabljati brez take povezave. Uporabniki mobilnih naprav potrebujejo povezavo do omrežja WI-FI ali pa morajo imeti zakupljen prenos podatkov pri njihovem ponudniku MNO.

Oddaljen varnostni element pomeni tudi zakasnitve s strani omrežja. Fizične imlementacij pošiljajo podatke nekaj centimetrov daleč, varnostni element v oblaku pa preko internetnega omrežja.

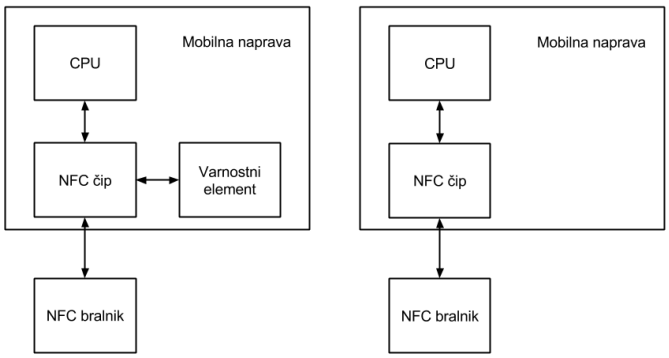
3.4.3 Primer

SimplyTapp je podjetje, ki je razvilo varnostni element v oblaku. Razvili so knjižnjice in spletno aplikacijo. V mobilni aplikaciji, ki želi uporabljati varnostni element, mora razvijalec vključiti njihovo knjižnico za mobilne naprave. Knjižnica poskrbi, da se ukazi APDU posredujejo na njihove strežnike. Preko spletnega vmesnika potem določi ključe, začetno strukturo in kakšne podatke želi shranjevati. SimplyTapp poskrbi za varnost na vseh nivojih, zato uporabniku za to ni potrebno skrbeti. Uporaba je močno poenostavljena.

3.5 Host Card Emulation

HCE je kratica za host card emulation, prevedeno posnemanje kartice na uporabnikovi napravi. HCE pravzaprav ni primer varnostnega elementa, saj je to aplikativna rešitev, ki ne vsebuje varnostnega elementa. Ta princip simulira varnostni element v procesorju naprave in komunicira direktno s čipom NFC, ki posreduje ukaze, namenjene bralniku.

Ker model HCE nima varnostnega elementa, je zelo redko, da se ga uporablja kot samostojno enoto. Pogosto se HCE uporablja v povezavi z oblakom.



Slika 3.4: Levo navaden model z varnostnim elementom, desno HCE model

Najpogostejše rešitve uporabljejo sistem začasnih virtualnih kartic. Ko ima mobilna naprava internetno omrežje, se sistem poveže na strežnik in generira začasno virtualno kartico ter jo prenese na uporabnikovo mobilno napravo v pomnilnik, ki ni varen. Za takšno delovanje so potrebne spremembe tudi na strani bralnikov, ki morajo imeti seznam teh začasnih virtualnih kartic.

3.5.1 Prednosti

Prednost HCE je v tem, da lahko tak način uporabljamo v katerikoli mobilni napravi. Druga prednost je ta, da je programiranje na napravi veliko lažje.

3.5.2 Slabosti

HCE nima varnostnega elementa, kar pomeni, da se emulacija kartic ne izvaja v izoliranem okolju. To pomeni, da ni mogoče zagotoviti visok nivo varnosti.

Ker naš model zahteva visok nivo varnosti, pomeni, da je potrebno uporabiti rešitev začasnih navideznih kartic. To pa pomeni spremembe na obstoječi infrastrukturi bralnikov, čemur pa se želimo izogniti.

3.5.3 Primer

Google je v KitKat različico Android operacijskega sistema vgradil podporo za HCE [9]. Razvojna knjižnica je namenjena razvijalcem. Delovanje v operacijskem sistemu je skladno s standardom ISO/IEC 14443-4. Operacijski sistem poskrbi za komunikacijo z anteno NFC in bralnikom. Razvijalcu za to ni potrebno skrbeti. Njegova naloga je le, da definira, za katere naslove AID želi, da operacijski sistem posreduje njegovi mobilni aplikaciji. To definira v menifestacijski datoteki aplikacije, kjer za izbrane naslove AID določi, objektom katerega razredu naj jih posreduje.

Ta razred mora razširiti abstrakten razred HostApduService in implemenirati dve metodi. Prva je onDeactivated(int reason), druga pa byte[] processCommandApdu(byte[] commandApdu, Bundle extras). Metoda onDeactivated se sproži takrat, ko se mobilna naprava odmakne od bralnika NFC, ali pa je bil izbran naslov AID, za katerega ne skrbi ta aplikacija. Druga metoda pa poskrbi za ukaze APDU, ki jih bralnik pošlje mobilni napravi. Razvijalcu je potem prepuščeno, da jih obdela lokalno oziroma pošlje v oblak ali kako drugo napravo.

```
public class MyHostApduService extends HostApduService {
    @Override
    public byte[] processCommandApdu(byte[] apdu, Bundle extras) {
        ...
    }
    @Override
    public void onDeactivated(int reason) {
        ...
    }
}
```

Slika 3.5: Nalepka kot varnostni element

3.6 Druge implementacije

Poleg standardnih oblik varnostnih elementov obstajajo tudi druge možnosti. Te zahtevajo dodatno strojno opremo. Varnotni element je lahko v obliki nalepke, v obliki zunanjega ohišja mobilnih naprav ali pa v dodatni strojni opremi, ki jo priklopimo preko vhoda USB ali vhoda za slušalke.

3.6.1 Prednosti

Prednost fizičnih dodatkov je ta, da je tudi tu dostop popolnoma pod nadzorom prodajalca, oz. izdajatelja dodatkov in ni odvisen od MNO, TSM in drugih posrednikov. Če dodatek izdamo sami, to pomeni, da imamo povsem proste roke glede tega, kaj se lahko izvaja na njem.

3.6.2 Slabosti

Podobno kot pri kartici MicroSD, je tudi tu problem to, da mora uporabnik dodatek kupiti ali pa mu ga moramo posredovati mi. MicroSD zahteva dodatno režo na mobilni napravi. Varnostni elementi v obliki zunanjih dodatkov pa spreminjajo izgled in velikost mobilne naprave. Uporabniki radi uporabljajo svoja ohišja, dodatki v obliki nalepk ali dodatnih priklopljenih naprav kvarijo izgled.

3.6.3 Primer

Twinee podjetja Twinlinx je varnostni element na nalepki. Nalepimo jo na hrbtni del mobilne naprave. Z njo komuniciramo direktno preko bralnika ali pa preko mobilne naprave in vgrajene antene NFC.

Poglavje 4

Težave in rešitev

4.1 Izbira varnostnega elementa

V prejšnjem poglavju so predstavljeni tipi varnostnih elementov s svojimi prednostmi in slabostmi. Ob implementaciji našega modela se soočimo s problemom izbire. Na žalost se podjetja, ki na trgu prodajajo varnostne elemente, niso odločila za skupno izvedbo ali podporo. Proizvajalci mobilnih naprav in mobilni operaterji se po svoje odločajo, katere tipe varnostnih elementov bodo podprli.

Če za naš model izberemo MicroSD, se odpovemo velikemu naboru mobilnih naprav, ki reže zanje nima. Ob izbiri kartice SIM postanemo močno odvisni od mobilnih operaterjev. Do vdelanih varnostnih elementov, razen redkih izjem, dostopa nimamo, saj jih v mobilnih napravah ni, ali pa se proizvajalec odloči, da ga bo uporabljal za svoje potrebe. Edina možnost, ki še ostane, je varnostni element v oblaku, saj smo v tem primeru še najamanj odvisni od drugih. Velika omejitev v tem primeru je to, da postanemo povsem odvisni od internetne povezave. Uporabniki lahko imajo zakupljen prenos podatkov, vendar za to nimamo zagotovila. Tudi pri tistih, ki to opcijo imajo zakupljeno, nimamo garancije, da bo internetno omrežje povsod dosegljivo. Težava nastane tudi pri gostovanju v tujini. Zato moramo uporabniku v primeru izbire oblaka zagotoviti tudi internetno povezavo.

Pri izbiri varnostnega elementa moramo biti pozorni tudi na to, katere izvedbe pametnih kartic ta varnostni element podpira. Varnostni elementi, ki podpirajo JavaCard tehnologijo in oblak, so programibilni, ostali so tipično predpripravljeni na eno izvedbo kartic. Če nam je dovolj podpora ene izmed izvedb, je izbira lažja, saj samo poiščemo varnostni element, ki podpira to izvedbo kartice.

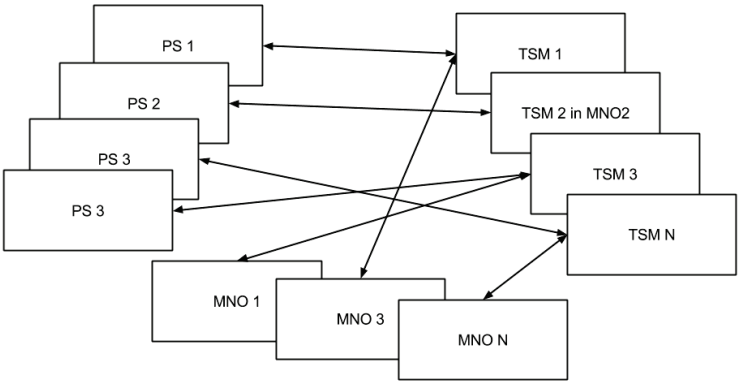
4.2 Kompleksnost MNO, TSM in ponudnikov storitev

Cilj modela je podpreti čimveč različnih ponudnikov storitev. V našem sistemu želimo gostiti navidezne pametne kartice različnih področij. To so kartice v javnem prevozu, sistemih garažnih hiš, kartice zvestobe, pa tudi plačilne kartice. Prav tako želimo, da se model uporablja v vseh državah sveta. Uporaba naj bo preprosta, pregledna in varna.

Število MNO je večkratnik števila držav na svetu, saj ima vsaka izmed njih vsaj enega, tipično pa več ponudnikov mobilnih storitev. Vsak ponudnik je samostojen in deluje na trgu, kjer se bori s konkurenco, zato sodelovanje med njimi ni prav pogosto.

Ponudnikov TSM je prav tako veliko. Ponudniki storitev so lahko sami svoj TSM, lahko pa za vlogo TSM najamejo zunanje izvajalce. Tudi MNO potrebuje vlogo TSM. Rabi jo za nadzor nad karticami SIM, ki jih izdaja svojim uporabnikom.

Zahteva, da so na mobilni napravi brestične pametne kartice različnih ponudnikov storitev, naredi naš model zelo kompleksen. N-ponudnikov lahko pomeni N-različnih vlog TSM. Če uporabimo najbolj razširjen varnostni element, kartico SIM, moramo v model dodati še M-različnih MNO.



Slika 4.1: Nepregleden model MNO, TSM in SP

4.2.1 Cena TSM in MNO

V našem poslovnem interesu je, da bi naš model omogočil gostovanje velikemu številu navideznih kartic. Poslovni interes leži v gostovanju navideznih kartic, saj za določeno ceno našo storitev nudimo ponudnikom storitev. Vsaka navidezna kartica zahteva sodelovanje s ponudnikom storitve. Takšno sodelovanje je močno zaželjeno.

Želimo pa se izogibati posrednikom TSM, saj je sodelovanje z njimi po navadi zelo drago, saj zahteva plačevanje dragih licenc in s provizijami pri transakcijah odžira zajeten del prihodkov. Nekateri ponudniki storitev imajo z njimi pogodbo o sodelovanju. Skrbijo za izdajo njihovih fizičnih kartic. Zamenjava njihovih fizičnih kartic z našimi virtualnimi za njih pomeni manjšo prodajo. Kar seveda ni v njihovem poslovnem interesu, zato bo sodelovanje z njimi zelo težko.

Prav tako se hočemo izogniti vlogam MNO in njhovim varnostnim elementom v kartici SIM. Ne toliko zaradi dragih pogodb in provizij, saj po navadi dostop omogočijo zastonj. Večja težava nastopi pri dostopu samem, saj mobilnim operaterjem pogosto ni v interesu omogočanje dostopa in spreminjanje njihovih sistemov zato, da bi pomagali nekemu drugemu podjetju.

4.3 Zagotavljanje varnosti

Pametne kartice pošiljajo ukaze, ki so kodirani. S tem zagotovimo, da napad prestreganja podatkov ni mogoč, saj napadalec ne pozna simetričnega ključa, ki si ga varnostni element in izdajatelj kartice delita.

Kartica takoj po izdelavi vsebuje privzeti ključ. Privzeti ključ je ponavadi sestavljen iz preprostega niza. Po navadi so to same ničle ali enice. Ker tak ključ ni varen, mora izdajatelj pred izdajo kartice uporabniku nanjo naložiti neki drug, bolj varen ključ. Ta proces imenujemo inicializacija. Izvede jo TSM, ponudnik storitev ali pa proizvajalec kartic med proizvodnjo.

Pri modelu, ki zamenja pametno kartico z mobilno napravo, se pojavi problem. Ključ, ki je poznan izdajatelju, varnostnemu elementu na telefonu ni znan. Zato je potrebno pred prvo uporabo kartice poskrbeti, da se inicializacija, ki jo izdajatelj izvede na fizični kartici, zgodi tudi na navidezni kartici na varnostnem elementu.

4.3.1 Algoritmi za enkripcijo

Večina varnostnih elementov v obtoku podpira enkripcijo DES. Algoritem je bil razvit s strani ameriške vlade v sedemdesetih letih. Zaradi kratkega ključa in modernejših računalnikov, ki so precej hitrejši kot v sedemdesetih, se ta način enkripcije smatra kot algoritem, ki ni varen, saj je ključ mogoče razbiti v manj kot enem dnevu. Algoritem 3DES kot direktna zamenjava naj bi bil dovolj varen. DES že od nastanka časa spremlja slab glas, saj mnogi verjamejo, da ima ameriška agencija NSA v algoritem vgrajene mehanizme, ki bi jim omogočili ekskluziven dostop. Priporočeno je, da se uporablja algoritem AES. Podpira ga velika večina varnostnih elementov, odprtokodni Cipurse pa gre še korak dlje in podpira samo AES.

DES ostaja težava, saj veliko varnostnih elementov podpira ta način enkripcije. Zato je od tistega, ki ima v ekosistemu NFC vlogo TSM odvisno, ali varnostni element dovolj varno uporablja ali ne. Ker v našem sistemu porpiramo več vrst brezstičnih pametnih kartic, moramo za tak primere ustrezno

poskrbeti. Najpogosteje se takšne napade preprečuje z omejenim številom napačnih poiskusov v določenem časovnem obdobju.

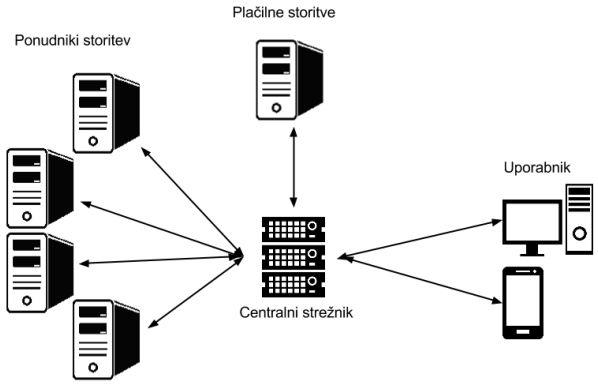
4.3.2 Izmenjava ključev

Da bi ključ izdajatelja kartice postavili na varnostni element, moramo vzpostaviti varen sistem izmenjave ključev. Prenos ključa na varnostni element moramo na neki način zakodirati. Ker to nikakor ne moremo storiti z izdajateljevim ključem, moramo uporabiti neki drugi ključ. Ključ, ki se nahaja na varnostnem elementu samem in ni znan nikomur, razen upravljavcu varnostnega elementa, kar je tipično TSM ali pa MNO, odvisno od tipa varnostnega elementa. Vzpostaviti je potrebno varni kanal OTA.

4.4 Predlagana rešitev: Centralizacija storitev

Glavni cilj je podpreti čimveč ponudnikov storitev. Pri tem se želimo izogniti posrednikom, kot so TSM, MNO in drugim. Te vloge sistem ne samo podraži, ampak tudi zaradi preprek onemogoči. Zato v sistem uvedemo centralni strežnik, ki je edina vstopna točka za uporabnika. Strežnik potem skrbi za ustrezno komunikacijo z različnimi ponudniki storitev.

Uporabnik lahko stanje svojih virtualnih kartic na telefonu ureja na več načinov. Prvi je preko spletne aplikacije, kjer na osebnem računalnika naroči novo kartico ali pa spreminja stanje na njej. Drugi način je, da to počne direktno iz mobilne aplikacije. Mobilna aplikacija ves čas v ozadju čaka na internetno povezavo in v trenutku, ko jo ima, začne z vzpostavitvijo varnega kanala in vzpostavljanju pravilnega stanja na varnostnem elementu v telefonu.



Slika 4.2: Centralizacija storitev

4.4.1 Centralni strežnik

Centralni strežnik je, kot že ime samo pove, osrednji element sistema. Vsa komunikacija poteka preko njega.

Uporabnik do podatkov dostopa preko spletnih servisev. Servise lahko kličejo različni odjemalci. Najpogostejša sta dva pristopa: preko spletne strani, kjer uporabnik preko osebnega računalnika kupuje nove vozovnice, ali pa preko mobilne aplikacije. Preko obeh naprav lahko uporabnik kupuje vozovnice, samo preko mobilne naprave pa lahko prenaša tudi kriptirane podatke, ki se kasneje zapišejo na varnostni element ali pa preko oblaka neposredno na mobilno napravo. Kriptirani podatki so ukazi APDU, saj so le ti tisti, ki jih varnostni element sprejema.

Komunikacija s spletnimi storitvami je za uporabnika skrita in se dogaja v ozadju, naloga aplikacij pa je, da prenešene podatke ustrezno prikaže v uporabniškem vmesniku.

Centralni strežnik s ponudniki storitev komunicira preko njihovih spletnih storitev ali kakega drugega protokola. Nekateri ponudniki take storitve že imajo. Edino, kar morajo dodatno podpreti, so protokoli za izmenjavo ukazov APDU.

Da bi lahko podprli vso komunikacijo uporabnika s strežnikom, moramo

v grobem podpreti naslednje spletne servise.

kreirajNovoKartico

Uporabnik preko tega servisa ustvari novo kartico. V njem pošlje, za katerega ponudnika storitev bi rad imel kartico in podatke, ki jih ponudnik storitve potrebuje. Po navadi so to ime, priimek in drugi podatki o uporabniku.

kupiStoritev

S tem servisom uporabnik kupuje nove storitve. Primer v javnem prevozu bi bil nakup nove vozovnice. Uporabnik preko spletne ali mobilne aplikacije izbere, kakšno karto bi rad kupil. Izbere lahko dan in uro, vhodno in izhodno postajo ter druge podatke. Centralni strežnik pri ponudniku storitev preveri, ali je karto mogoče kupiti. Uporabniku posreduje odgovor in morebitne druge možnosti.

placajInPotrdiStoritev

V tem servisu dobimo potrditev s strani uporabnika. Prav tako pridobimo podatke, ki so potrebni pri plačilu. Uporabnik ima po tem, ko je plačilo uspešno izvedeno, karto kupljeno.

apduIzmenjava

Uporaba tega servisa je odvisna od tega, za kakšen tip varnostnega elementa gre. Če ima uporabnik varnostni element na mobilni napravi, potem se ta servis uporabi za prenos podatkov o karti na ta varnostni element.

Če pa gre za varnostni element v oblaku, pa se ta servis uporabi za izmenjavo ukazov APDU med bralniki in strežnikom, kjer se nahaja varnostni element.

4.4.2 Varnostni element

Centralni sistem je dokaj neodvisen od tega, kakšen varnostni element izberemo. Je pa to še vedno zelo pomembni del sistema in mu moramo posvetiti poznornost.

Varnostni element je zaradi nestandardnosti trenutno največja šibka točka sistema, saj je potrebno zaradi pomanjkljivosti standardnih izvedb narediti odločitev, ki ni optimalna. Ker želimo ohraniti neodvisnost od posrednikov, smo trenutno prisiljeni za varnostni element izbrati JavaCard in kartico MicroSD. Predvsem zaradi programibilnosti in prilagajanja različnim izvedbam pametnih kartic.

Ker pa mobilni svet postaja bolj povezljiv iz dneva v dan, predvidevamo, da bo model v prihodnosti za varnosti element uporabljal rešitev v oblaku. Predstavljeni bosta ob rešitvi.

4.4.3 Rešitev z MicroSD

Najboljša izbira v prehodnem času je izbira MicroSD. Na trgu obstaja več različnih modelov. Najprimernejši je tisti, ki je zmožen poganjati JavaCard. Tako lahko na njem napišemo poljuben applet, ki podpira vse vrste varnostnih elementov, kot so Cipurse, Mifare linija, Felica, Calipso in podobni. To so namreč najbolj razširjene pametne kartice na trgu in jih je zato tudi najbolj smiselno podpreti. Še posebaj, če upoštevamo obstoječo infrastrukturo bralnikov.

Uporabniku, ki želi uporabljati naš model, moramo zagotoviti, da bo do kartice MicroSD lahko prišel. Ker naš model implementira naše podjetje, smo mi tisti, ki moramo uporabnikom izdati novo kartico MicroSD. Dodaten strošek bi uporabnike lahko odvrnil od uporabljanja storitve, zato moramo premisliti tudi o delni ali polni subvenciji kartice MicroSD. Prav tako je pred uporabo nanjo potrebno naložiti aplete JavaCard in začetne varnostne ključe. Dostop je v vsakem primeru mogoč samo nam, izvajalcu storitve.

Mnogoličnost pri MicroSD

Pametne kartice novejših tehnologij privzeto omogočajo, da je na njej lahko zapisanih več različnih skupin podatkov, ki so izolirane. Na fizični pametni kartici bi tako lahko na primer imeli podatke več različnih bank in računov. Vendar se to v praksi ne dogaja, saj vsaka banka tipično izda samostojno kartico, nanjo pa ne pusti pisati drugih podatkov.

V našem modelu želimo uporabljati več virtualnih kartic. Ker je cilj ohraniti trenutno arhitekturo pametnih kartic, moramo podpreti možnost dodajanja celotne navidezne kartice. To imenujemo mnogoličnost, saj imamo hkrati na varnostnem elementu implementiranih več virtualnih kartic.

Zaradi različnih izvedb kartic (Mifare, Calypso, Cipurse ...) pa moramo poskrbeti tudi različne izvedbe virtualnih kartic. Za vsako izvedbo kartice moramo napisati novi applet, ki skrbi za določeno izvedbo. Za krmiljenje med njimi potrbujemo še en applet, ki poskrbi za izbiro ustreznega appleta izvedb. Imenujemo ga vhodni applet.

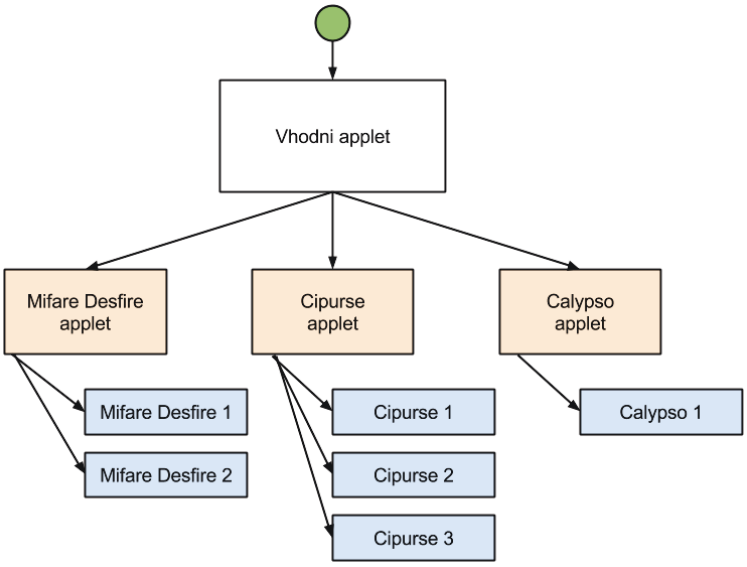
Na sliki 4.3 je prikazana arhitektura na varnostnem elementu. Vsak dostop do varnostnega elementa gre preko vhodnega appleta. Uporabnik mu preko aplikacije na mobilnem telefonu pove, katero kartico želi uporabljati. Vhodni applet potem poskrbi, da izbere pravilni applet in navidezno kartico. Dokler je izbrana določena navidezna kartica, bodo vsi ukazi, ki pridejo iz bralnikov NFC, posredovani direktno do nje. Ker je ključ navidezne kartice shranjen na izvedbi navidezne kartice, je dostop do podatkov omogočen samo izdajatelju navidezne kartice.

Vhodni applet

Za upravljanje z navideznimi karticami potrebujemo vhodnemu appletu definirati ukaze. Tabela 4.1 prikazuje seznam potrebnih ukazov.

Spodaj je primer kode metode vhodnega appleta, ki ustvari novo navidezno kartico. Vhodni applet podpira 4 različne izvedbe pametnih kartic.

```
private void createCard(APDU apdu, byte[] buffer) {
```



Slika 4.3: Mnogoličnost

Ukaz	Razlaga
E1	začetek vzpostavitve sejnega ključa
EC	nadaljevanje vzpostavitve SK
EA	ustvari novo virtualno kartico
E5	aktiviraj kartico
E0	deaktiviraj kartico
EF	pridobitev seznam navideznih kartic
E6	pridobi uporabnika
ED	izbriši kartico
E7	spremeni pin uporabnika

Tabela 4.1: Ukazi vhodnega appleta

```
if (this.numberOfcards == MAX_CARDS) {
ISOException.throwIt(Util.ENTRYAPPLET_BOUNDARY_ERROR);
}

if ((buffer[ISO7816.OFFSET_LC]) != 17) {
ISOException.throwIt(Util.ENTRYAPPLET_LENGTH_ERROR);
}

byte id = buffer[ISO7816.OFFSET_CDATA];

for (byte i = 0; i < this.numberOfcards; i++) {
if (this.cardIDs[i] == id) {
ISOException.throwIt(Util.ENTRYAPPLET_DUPLICATE_ERROR);
}
}

byte type = buffer[ISO7816.OFFSET_P1];

this.cardIDs[this.numberOfcards] = id;
if (type == 0) {
this.cards[this.numberOfcards] = new DesfireVirtualCard();
} else if (type == 1) {
this.cards[this.numberOfcards] = new ClassicVirtualCard();
} else if (type == 2) {
this.cards[this.numberOfcards] = new CipurseVirtualCard();
} else if (type == 3) {
this.cards[this.numberOfcards] = new CalypsoVirtualCard();
} else {
ISOException.throwIt(ISO7816.SW_INCORRECT_P1P2);
}
```

```
this.numberOfcards++;

// Persistent, ker se pošlje kot parameter
byte[] decryptedMasterKey = new byte[KEY_LENGTH];

// kriptiran masterKey je v polju za podatke od APDU
sessionKeyDecipher.doFinal(buffer, (short) (ISO7816.OFFSET_CDATA + 1),
KEY_LENGTH, decryptedMasterKey, (short) 0);

// byte[] decryptedMasterKey = ;

// Call changeKey
this.cards[this.numberOfcards - 1]
.changeMasterKey(decryptedMasterKey);
}
```

Applet Desfire

Za vsako izvedbo pametne kartice, ki jo želimo podpreti v našem sistemu, moramo napisati novi applet, ki jo bo simuliral. Za primer bo uporabljena Mifare Desfire izvedba, ki je med najbolj razširjenimi[10]. Ostale izvedbe kartic, kot so Cipurse, Calypso in druge, imajo podobno implementacijo. Razlikujejo se po navadi v tipih enkripcije in oblikah ukaza.

Spodaj je primer kode appleta Desfire, ki prejme ukaz APDU in ga posreduje metodam, ki so zadolžene za določen ukaz.

```
public void process(APDU apdu) {

if (selectingApplet()) {
clear();
return;
}
```

```
byte[] buffer = apdu.getBuffer();

if (authenticated == -1)
this.securityLevel = Util.PLAIN_COMMUNICATION;
if ((commandToContinue != Util.NO_COMMAND_TO_CONTINUE)
&& (buffer[ISO7816.OFFSET_INS] != Util.CONTINUE)) {
clear();
ISOException.throwIt((short) Util.COMMAND_ABORTED);
}

// check the INS byte to decide which service method to call
switch (buffer[ISO7816.OFFSET_INS]) {
case Util.AUTHENTICATE:
authenticate(apdu, buffer, (byte) 0, (short) 8);
break;
case Util.AUTHENTICATE_ISO:
authenticate(apdu, buffer, (byte) 1, (short) 8);
break;
case Util.AUTHENTICATE_AES:
authenticate(apdu, buffer, (byte) 2, (short) 16);
break;
case Util.CHANGE_KEY_SETTINGS:
changeKeySettings(apdu, buffer);
break;
case Util.CHANGE_KEY:
changeKey(apdu, buffer);
break;
case Util.CREATE_APPLICATION:
createApplication(apdu, buffer);
break;
```

```
case Util.DELETE_APPLICATION:
deleteApplication(apdu, buffer);
break;
case Util.GET_APPLICATION_IDS:
getApplicationIDs(apdu, buffer);
break;
case Util.GET_KEY_SETTINGS:
getKeySettings(apdu, buffer);
break;
case Util.SELECT_APPLICATION:
selectApplication(apdu, buffer);
break;
case Util.FORMAT_PICC:
formatPICC(apdu, buffer);
break;
case Util.SET_CONFIGURATION:
setConfiguration(apdu, buffer);
break;
case Util.GET_FILE_IDS:
getFileIDs(apdu, buffer);
break;
case Util.CREATE_STDDATAFILE:
createStdDataFile(apdu, buffer);
break;
case Util.CREATE_BACKUPDATAFILE:
createBackupDataFile(apdu, buffer);
break;
case Util.CREATE_VALUE_FILE:
createValueFile(apdu, buffer);
break;
case Util.CREATE_LINEAR_RECORD_FILE:
```

```
createLinearRecordFile(apdu, buffer);
break;
case Util.CREATE_CYCLIC_RECORD_FILE:
createCyclicRecordFile(apdu, buffer);
break;
case Util.DELETE_FILE:
deleteFile(apdu, buffer);
break;
case Util.READ_DATA:
readData(apdu, buffer);
break;
case Util.WRITE_DATA:
writeData(apdu, buffer);
break;
case Util.GET_VALUE:
getValue(apdu, buffer);
break;
case Util.CREDIT:
credit(apdu, buffer);
break;
case Util.DEBIT:
debit(apdu, buffer);
break;
case Util.READ_RECORDS:
readRecords(apdu, buffer);
break;
case Util.WRITE_RECORD:
writeRecord(apdu, buffer);
break;
case Util.CLEAR_RECORD_FILE:
clearRecordFile(apdu, buffer);
```

```
break;
case Util.COMMIT_TRANSACTION:
commitTransaction(apdu, buffer);
break;
case Util.ABORT_TRANSACTION:
abortTransaction(apdu, buffer);
break;
case Util.CONTINUE:
switch (commandToContinue) {
case Util.AUTHENTICATE:
authenticate(apdu, buffer, (byte) 0, (short) 8);
break;
case Util.AUTHENTICATE_ISO:
authenticate(apdu, buffer, (byte) 1, (short) 8);
break;
case Util.AUTHENTICATE_AES:
authenticate(apdu, buffer, (byte) 2, (short) 16);
break;
case Util.GET_APPLICATION_IDS:
getApplicationIDs(apdu, buffer);
break;
case Util.READ_DATA:
sendBlockResponse(apdu, buffer, dataBuffer, offset,
fileSecurityLevel);
break;
case Util.WRITE_DATA:
writeData(apdu, buffer);
break;
case Util.READ_RECORDS:
readRecords(apdu, buffer);
break;
```

```
case Util.WRITE_RECORD:
writeRecord(apdu, buffer);
break;
default:
ISOException.throwIt(Util.ILLEGAL_COMMAND_CODE);
break;
}
break;
default:
ISOException.throwIt(Util.ILLEGAL_COMMAND_CODE);
break;
}

}
```

Primer pisanja na Desfire navidezno kartico

Komunikacija poteka med navadnim bralnikom podjetja, ki izvaja storitve, in varnostnim elementom na MicroSD kartici. V primeru javnega prevoza bi to bil bralnik na avtobusni ali železniški postaji. Prvi par APDU je namenjen vhodnemu appletu na MicroSD, vsi ukazi, ki mu sledijo, pa so namenjeni appletu Desfire. Primer zaporedja ukazov APDU je prikazan na sliki 4.4. Na navidezno kartico izvedbe Desfire želimo zapisati krajši niz. Primer bi bil podaljšanje veljavnosti mesečne vozovnice. Vsi ukazi so za lažje branje predstavljeni v šestnajstiški obliki.

1. par APDU je namenjen vhodnemu appletu. Z njim mobilna aplikacija določi, kateri navidezni kartici bodo namenjeni ukazi, ki mu sledijo. Vsi ukazi APDU, ki so namenjeni vhodnemu appletu, se začnejo z "EA"(šestnajstiško), kar je okrajšava za angleški prevod Entry Applet. Vse te ukaze smo izbrali mi in ustrezajo ISO/IEC 14443 standardu. Drugi bajt po standardu določi tip ukaza. "E5"je določen za izbiro kartice. Bajta "12 34"predstavljata id kartice. Id se določi pri kreiranju nove kartice.

Vhodni applet odgovori z "EA 00", kar pomeni, da je bil ukaz izbire uspešno izveden. V tem trenutku je navidezna kartica izbrana in pripravljena na prejemanje ukazov s strani bralnika. Uporabnik mobilno napravo približa bralniku, ki začne s pošiljanjem ukazov. Vhodni applet bo vse ukaze, ki se ne začnejo z EA, samo posredoval zadnjemu izbranemu appletu in kartici.

2. par APDU je namenjen appletu, ki skrbi za Desfire izvedbe navideznih kartic. Podjetje, ki izvaja storitve preko bralnika, najprej pošlje ukaz "5A", ki zahteva izbero aplikacije. Aplikacija je izraz, ki ga Mifare v Desfire izvedbi uporablja za podimenike. Vsaka kartica ima lahko več podimenikov, vsak podimenik pa več datotek, v katere je mogoče pisati podatke. Bajti "33 44 55"predstavljajo AID podimenika.

V primeru uspešne izvedbe applet Desfire odgovori z "91 00", kar predstavlja uspešno izvedbo ukaza.

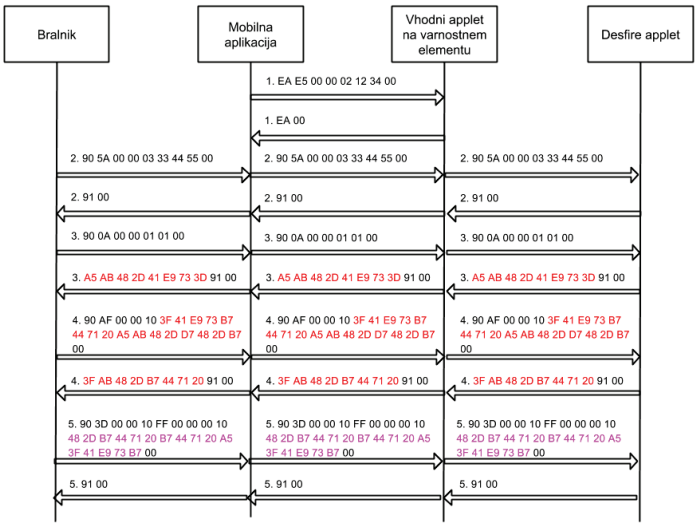
3. par APDU začne avtentikacijo in vzpostavljanje sejnega ključa. Ukaz pravi, da naj se za šifriranje uporabi ključ številka 1 (01) in šifrirni algoritem DES ("0A").

Odgovor vsebuje 8 bajtov šifriranega naključnega niza, imenovanega naključni B. Bajti, pobarvani rdeče, predstavljajo podatke šifrirane z glavnim ključem podimenika.

4. par APDU nadaljuje z vzpostavitvijo sejnega ključa. V ukazu sta šifrirana naključna niza A in B. "AF"po Desfire specifikacijah pomeni nadaljevanje prejšnjega ukaza.

Applet Desfire v primeru uspešnega preverjanja naključnega niza B odgovori s prejetim nizom A. Ko bralnik prejme in uspešno preveri niz A, se vzpostavi sejni ključ. To je začasni ključ, ki se uporablja pri naslednjih ukazih. Na sliki so bajti, šifrirani s tem sejnim ključem, obarvani vijolično. Postopek avtentikacije je sedaj zaključen.

5. par APDU je namenjen pisanju. Bajt "3D"pomeni pisanje. Bajt "FF"predstavlja datoteko v katero želimo pisati. Bajti "00 00 00"predstavljajo odmik. To je kje v datoteki želimo pisati. Same ničle pomenijo začetek datoteke. Bajti, obarvani vijolično, so šifrirani podatki, ki jih želimo zapisati.



Slika 4.4: Izmenjava ukazov APDU za pisanje

Applet Desfire s sejnim ključem dešifrira niz, ga zapiše in odgovori z uspešno izvedbijo.

Zagotovitev varnosti

Pametne kartice so v času izdaje uporabniku v tako imenovanem inicializiranem stanju. V tem stanju so nanjo naloženi ključi in začetno stanje hirearhije datotek na njej. V našem primeru, ko imamo navidezne pametne kartice, moramo enako začetno stanje vzpostaviti tudi v varnostnem elementu.

Prazna navidezna kartica ima ob tem, ko naredimo novo instanco, privzete ključe. Ti so za vsak tip pametnih kartic znani. To pomeni potencialno varnostno luknjo, saj bi napadalec lahko pri inicializaciji spremljal promet med bralnikom in varnostnim elementom ter tako dobil varnostni ključ. Ko bi tega imel, bi lahko poljubno spreminjal vsebino kartice. Zato moramo v času prenosa začetnega ključa na varnostni element poskrbeti za varni kanal.

To naredimo tako, da za inicializacijo kartice zadolžimo vhodni applet. Vhodni applet je v času izdaje že naložen na varnostnem elementu. Ima svoj ključ, s katerim lahko zagotovi vzpostavitev varnega kanala. Uporabimo

enkripcijo AES in vzpostavitev sejnega ključa, s katerim potem šifriramo začetni ključ virtualne kartice.

Ko je ključ instance virtualne kartice varno prenešen v varnostni element, lahko varno izvajamo ostale ukaze, ki jih izdajatelj pametne kartice uporablja.

Na ta način zagotovimo varnost izdajatelja pametne kartice, saj uporabniki ne morejo sami spremnijati vsebino virtualne kartice. Ni pa s tem poskrbljeno za varnost uporabnika. Napadalec bi lahko po kraji uporabljal njegov varnostni element. Lahko bi poskušal tudi prebrati vsebino s svojim prenosnim bralnikom. Da to preprečimo, damo uporabniku možnost uporabe številke PIN. To lahko uporablja ob vsaki pomembnejši interakciji z varnostnim elementom, ki spreminja vsebino. Ob vnosu številke PIN za krajši čas omogočimo dostop do varnostnega elementa.

OTA za MicroSD

Na sliki 4.5 je prikazan proces vzpostavitve nove virtualne kartice. V trenutku, ko uporabnik želi uporabljati novo navidezno pametno kartico, mobilna aplikacija sproži zahtevo za incializacijo sejnega ključa. Vhodnemu appletu na varnostnem elementu pošlje ukaz APDU za začetek inicializacije sejnega ključa (1. korak).

Vhodni applet na varnostnemu elementu generira naključni niz B. Z glavnim ključem vhodnega appleta ta niz šifrira in ga pošlje v odgovoru APDU. Mobilna aplikacija APDU samo posreduje strežniku (2.korak).

Strežnik z istim ključem kot ga ima vhodni applet naključni niz odšifrira, doda naključni niz A, vse skupaj šifrira in pošlje preko mobilne aplikacije v novem ukazu proti vhodnemu appletu na varnostnem elementu (3. korak).

Varnostni element preveri pravilnost odšifriranega naključnega niza B. Če je niz pravilen generira sejni ključ in odgovori s šifriranim naključnim nizom A. Mobilna aplikacija zopet samo posreduje ukaz strežniku. Strežnik preveri pravilnost naključnega niza A in prav tako generira sejni ključ v primeru, da se ujema s poslanim. Varnosti element in strežnik si sedaj delita sejni ključ, s katerim lahko varno šifrirata občutljive podatke (4. korak).

Strežnik mora sedaj pridobiti novi ključ za navidezno kartico. Ta ključ mora biti znan tudi ponudniku storitev, ki bo kasneje preko svojih bralnikov pošiljal šifrirane ukaze. Zato preko povezave SSL ponudniku storitev pošljemo zahtevo za novi ključ (5. in 6. korak).

Ko strežnik pridobi ključ, ga mora posredovati varnostnemu elementu. Pošlje mu ukaz APDU za kreiranje nove kartice. V njem poda podatke o tem, kakšno izvedbo kartice želi, da vhodni applet kreira. V istem ukazu poda tudi ključ in naslov AID, s katerim bo v prihodnje naslavljal virtualno kartico. Strežnik del ukaza s ključem šifrira in pošlje na varnostni element (7. korak).

Varnostni element odšifrira ključ iz ukaza APDU in generira željeno novo instanco. Po uspešnem kreiranju nove virtualne kartice pošlje potrditev. To potrditev mobilna aplikacija posreduje strežniku, ta pa naprej ponudniku storitve. Sedaj vsi v verigi vedo, da je bila kreirana nova virtualna kartica, ki je pripravljena na uporabo (8. korak).

Ukazi in odgovori APDU, ki so potrebni za vzpostavitev nove virtualne kartice:

EA E1 00 00 00 – sproži začetek inicializacije,

[16 šifriranih bajtov] EA 00 – odgovor z naključnim nizom B,

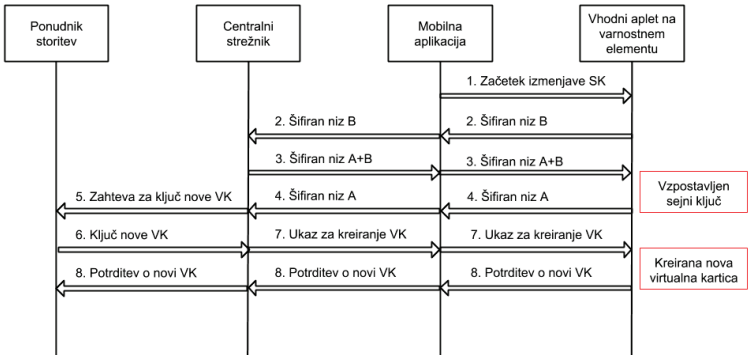
EA EC 00 00 [32 šifriranih bajtov] 00 - ukaz za nadaljevanje inicializacije z naključnima nizoma A in B,

[16 šifriranih bajtov naključnega niza A] EA 00 – odgovor z naključnim nizom A,

EA EA 01 00 11 05 [16 šifriranih bajtov] 00 - -ukaz za kreiranje nove virtualne kartice Bajt ”01”predstavlja tip Desfire ”05”pa id ponudnika storitev.

4.4.4 Rešitev v oblaku

Ko govorimo o varnostnem elementu v oblaku to pomeni, da nimamo pravega varnostnega elementa v fizični obliki, temveč se ta nahaja na strežnikih, preko katerih dostopamo do podatkov. V času uporabe virtualne kartice moramo poskrbeti, da ima mobilna naprava delujočo internetno povezavo,



Slika 4.5: OTA za MicroSD

saj sicer virtualne kartice ne moremo uporabljati. To pogosto zahteva poseg v strojno opremo, čemur pa se želimo izogniti. Na lokacijah, kjer uporabniki uporabljajo kartice, namreč brezžični WiFi internet pogosto ni dostopen. V tem primeru se moramo zanašati na internetno povezavo iz mobilnega omrežja. Tako postanemo odvisni od uporabnikovega mobilnega paketa.

Oddajnik NFC na mobilni napravi lahko poleg prenosa podatkov uporabimo tudi za vzpostavitev začasne internetne povezave. Ko uporabnik približa mobilno napravo bralniku, ta lahko pošlje potrebne informacije za vzpostavitev brezžične povezave z routerji v bližini. Po navadi je to ime routerja in geslo. Povezava je lahko začasna in deluje le v času, ko je uporabnik dovolj blizu bralniku. To preprečuje, da bi uporabniki uporabljali omrežje za druge vrste komunikacij in pri tem mašili ter upočasnili delovanje. Kot rečeno mora ponudnik storitev pri tem zagotoviti brezžično omrežje.

Strežniško kodo je veliko lažje vzdrževati in ob dodajanju novih tipov varnostnih protokolov ni potrebno spreminjati programov na drugih varnostnih elementih. Uporabniku v tem primeru ni potrebno prinašati mobilne naprave z varnostnim elementom do našega storitvenega centra, kjer bi mu naložili novo različico progama na varnostnem elementu.

Varnostni element v oblaku pomeni, da avtomatično zgubimo vse posrednike, ki so udeleženi pri dovoljenju dostopa do varnostnega elementa. Nismo odvisni od prozvajalcev, ki bi ali pa ne vgrajevali varnostnega elementa in

rež za kartice. Prav tako za SIM in druge dodatke.

Mnogoličnost v oblaku

Pri varnostnem elementu v oblaku prav tako potrebujemo podoben koncept vhodnega appleta, kot ga imamo pri MicroSD. Prav tako moramo napisati posebne razrede, ki so sposobni izmanjave ukazov APDU. Za vsako izvedbo moramo napisati knjižnico, ki to izvedbo simulira. Ker lahko programiramo v poljubnem jeziku, je razvoj hitrejši in lažji. Prav tako ni težav z omejitvijo pomnilnika. Do varnostnega elementa dostopamo preko spletne storitve.

Vhodni spletni servis

Za komunikacijo med mobilno napravo in varnostnim elementu ohranimo izmenjavo ukazov APDU, čeprav ni potrebna. Za upravljanje navideznih kartic uporabimo enak nabor ukazov, kot jih uporablja vhodni applet pri MicroSD, saj ni potrebe po spreminjanju. Prehodu iz MicroSD na varnostni element v oblaku je zaradi tega lažji, saj so potrebne manjše spremembe na mobilni napravi. Spremeniti je potrebno le tisti del kode, ki posreduje ukaze.

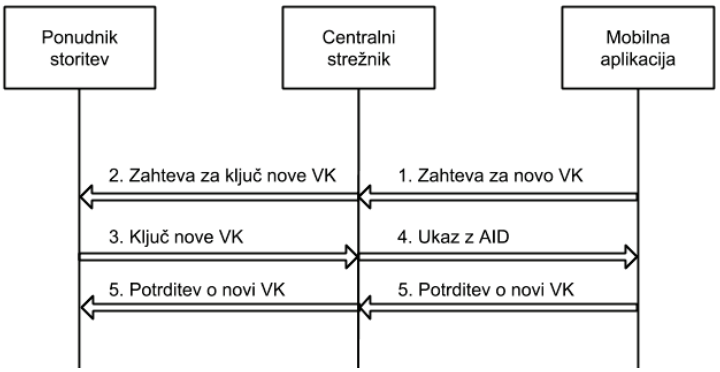
Zagotovitev varnosti in OTA v oblaku

Na sliki 4.6 je prikazan proces vzpostavitve nove virtualne kartice v oblaku. Ker je varnostni element kar na samem strežniku, je postopek kreiranja nove virtualne kartice veliko preprostejši.

Uporabnik v mobilni aplikaciji sproži postopek kreiranje nove navidezne kartice. Mobilna aplikacija strežniku pošlje zahtevo za novo navidezno kartico (1. korak).

Centralni strežnik prav tako kot pri OTA za MicroSD preko varne povezave SSL od ponudnika storitev pridobi ključ nove navidezne kartice (2. in 3. korak).

Pridobljeni ključ shrani na strežniku, mobilni aplikaciji pa le posreduje naslov AID, ki pripada novi kartici (4. korak).



Slika 4.6: OTA za oblak

Aplikacija strežniku potrdi kreiranje nove kartice, strežnik pa pošlje potrditev še ponudniku storitev (5. korak).

Poglavje 5

Zaključek

Brezstične pametne kartice so že začele nadomeščati obstoječe kartice čip in PIN. Implementacija navideznih kartic pa po hitrosti širitve zankrat še zaostaja. Večina rešitev, ki so že v uporabi, implementira eno samo navidezno kartico. Rešitve, ki bi podpirala več kartic različnih ponudnikov storitev na istem varnostnem elementu, še vedno ni v praktični uporabi.

V prihodnosti se bo trg verjetno razdelil na manjše segmente, vsak s svojim naborom kartic ponudnikov storitev. V idealnem scenariju bi se vzpostavil standard, ki bi omogočal gostovanje navidezne kartice v drugih sistemih. Če se bo to zgodilo, bomo videli v bližnji prihodnosti. Odvisno je predvsem od prvega vala implementacij, ki bodo začrtale smer razvoja. Predvidevamo, da bodo vedno večjo veljavo dobile programske rešitve. Hitra prilagodljivost in več izvedb odtehta prednosti rešitev v obliki strojne opreme. Edina prednost pri slednjih ostaja hitrost izvajanja, ki pa se zaradi namenskih čipov v varnostnih elementih in optimizacij hitro zmanjšuje.

Katera programska rešitev bo prevladala, je težko reči. Možno je, da bo določena vrsta varnostnih elementov prevladala in postala standard. Predvidevamo pa, da bo zaradi vedno boljše povezljivosti naprav v internetna omrežja, hitrejših prenosov in manjših latenc, ki jih prinaša 4G, na koncu prevladala rešitev v oblaku. Tej rešitvi smo posvečali manj pozornosti, saj je trenutno uporaba MicroSD bolj praktična. Bo pa v prihodnosti več truda

namenjenega ravno v varnostni element v oblaku in v rešitve, ki jih je z njimi mogoče narediti.

Literatura

[1] Standard ISO/IEC 7810. Dostopno na:
http://en.wikipedia.org/wiki/ISO/IEC_7810

[2] Standard ISO/IEC 7816. Dostopno na:
http://en.wikipedia.org/wiki/ISO/IEC_7816

[3] Standard ISO/IEC 14443. Dostopno na:
http://en.wikipedia.org/wiki/ISO/IEC_14443

[4] Prihod brezstičnih pametnih kartic v Slovenijo. Dostopno na:
http://www.delo.si/gospodarstvo/potrosnik/brezsticno-placevanje-prihaja-tudi-v-slovenijo.html

[5] Datotečna struktura po ISO 7816-4 je dostopna na:
http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816–4_5_basic_organizations.aspx

[6] Prodajni deleži pametnih kartic za potrebe prevoza. Dostopno na:
https://technology.ihs.com/425058/smart-cards-in-transportation-report-2013

[7] Ranljivost Mifare Classic. Dostopno na:
http://www.cs.ru.nl/ flaviog/publications/Attack.MIFARE.pdf

[8] JavaCard specifikacija. Dostopna na:
http://www.oracle.com/technetwork/java/javacard/overview/index-jsp-140503.html

[9] JavaCard specifikacija. Dostopna na:
https://developer.android.com/guide/topics/connectivity/nfc/hce.html

[10] Pametne kartice podjetja NXP. Dostopno na:
http://www.nxp.com/products/identification_and_security/smart_card_ics/